Preface

Thank you and congratulations for purchasing F-PROT Professional. It will protect your data from virus infections and give you peace of mind knowing that your valuable data is safe.

Information is often one of the most valuable assets a company has. New technology has enabled us to handle information in an unprecedented way, but it also presents new threats. In the 80's and 90's, an increasing portion of information processing has moved to workstations and personal computers. This has created new information security problems that must be addressed.

One of the most potentially harmful threats is viruses. The number of computer viruses has increased from tens to hundreds and thousands in just a few years. An alarming portion of viruses is deliberately programmed to destroy data — they're not young hackers' harmless pranks but a real threat that can cause irreparable damage if proper precautions are not taken.

Knowing how to avoid virus infections and having an efficient anti-virus program are two important steps in reducing the risk of infections and resulting data loss.

This manual is the user guide and reference manual for the users of F-PROT. The second half of the manual also contains background information on viruses in general and practical advice on how to protect computers against them.

Data Fellows Ltd. offers professional information security services and products. Our activities include software development, training and consulting.

F-PROT Professional, the technological leader in anti-virus software, is a fitting addition to our service and product assortment.

Data Fellows Ltd
F-PROT Support
Wavulintie 10
FIN-00210 Helsinki
FINLAND

☎ +358-0-692 3622
fax +358-0-670 156
email Data.Fellows@DF.elma.fi

1 F-PROT Quick Start 6
1.1 Installation of F-PROT 6
1.2 Manual Installation and Customization 6
1.3 Basic Procedures 7
1.4 The Components of the F-PROT Package 8
1.5 Minimum Configuration 9
2 Installation and Settings 10
2.1 Installation on the Hard Disk 10
2.2 Selecting the Language 11
2.3 Customizing the Message Displayed by VIRSTOP 12
2.4 Executing VIRSTOP at Startup 12
2.5 Command-line switches supported by VIRSTOP 12
2.5 VIRSTOP for Windows 14
2.5 Optimizing the Memory Usage of VIRSTOP 16
2.6 Other Settings 16
2.7 Installing F-PROT onto a File Server 17
2.8 Registering the Program 18
3 Scanning for Viruses 19
3.1 Quick Scan, Secure Scan and Heuristic Analysis 19
3.2 Selecting the Scanning Options 20
3.3 Starting the Scan 23
3.4 Disinfecting Files 26
3.5 Creating an Anti-Virus Diskette for Emergency Situations 27
4 VIRSTOP and F-TEST 30
4.1 Verifying the Functionality of VIRSTOP 31
5 Compatibility, Updates and Support 32
5.1 Updates 32
5.2 Support 32
5.3 Common Questions and Answers 33
5.4 Further Development of F-PROT 34

ADVANCED FEATURES

6 Heuristic Analysis 36
7 Command Line Mode 39
8 Using F-PROT under Windows and OS/2 42
9 Automating Scans under DOS with the F-AUTO 43

COMPUTER VIRUSES

10 What To Do When a Virus Strikes 46

11 Computer Viruses: Background Information 49

11.1 Viruses, Trojan Horse Programs, and Other Malicious Software 49

11.2 Situation in Scandinavia 52

11.3 Working Mechanisms of Viruses in MS-DOS 53

11.4 Advanced Methods Used by Viruses 57

Stealth Viruses 58

Self-Encrypting, Polymorphic and Mutation viruses 58

Retroviruses 60

Application Specific Viruses 60

Multipartition Viruses 61

11.5 How Great is the Virus Threat? 61

Case: Lehigh University 62

Case: AIDS Information Diskette 62

Case: Cascade Infection in Finland 63

11.6 Viruses in OS/2 63

11.7 Viruses in Windows 65

12 How to Use Your Computer Safely 69

Glossary 70

Index 79

Appendix 1: F-SCHEDULER

Appendix 2: F-CHECK

Appendix 3: Version Bulletins.....

Using F-PROT

1 F-PROT Quick Start

The Quick Start chapter is meant for the experienced user, who has installed and used similar applications before. Chapter 2 contains more detailed installation instructions. Although it is quite possible to use F-PROT after reading only the Quick Start, we recommend reading the whole manual, since it contains valuable information about F-PROT and computer viruses.

Chapter 10: "Computer Viruses — Background Information" is meant for those readers who want to know more about viruses in general. Chapter 10 describes common virus types and infection mechanisms as well as the steps that should be taken by persons responsible for protecting computers against virus attacks.

1.1 Installation of F-PROT

F-PROT is installed by starting F-PROT.EXE from a floppy disk and selecting **Install** from the **Install** menu.

The installation process copies all program files to a directory on the hard disk. The memory-resident virus scanning portion of the program, VIRSTOP, is localized for the selected language and can also be configured to display a customized message when a virus is encountered.

VIRSTOP can be installed for permanent use by selecting the command **VIRSTOP** from the **Install** menu. This writes the command for starting VIRSTOP as the first line of AUTOEXEC.BAT.

VIRSTOP will be activated during next startup. F-PROT can be executed with the command

```
C:\> F-PROT\F-PROT
```

1.2 Manual Installation and Customization

For manual installation, make sure that at least one megabyte of hard disk space is available. Create a new directory for F-PROT — C:\F-PROT is recommended. Copy everything on the distribution diskette into this directory.

If desired, the directory can be added to the DOS command path by adding its name to the PATH line of AUTOEXEC.BAT.

VIRSTOP is installed by adding the following line as the first line of AUTOEXEC.BAT

```
C:\F-PROT\VIRSTOP
```

or the following line as the last line of CONFIG.SYS:

```
DEVICE=C:\F-PROT\VIRSTOP.EXE
```

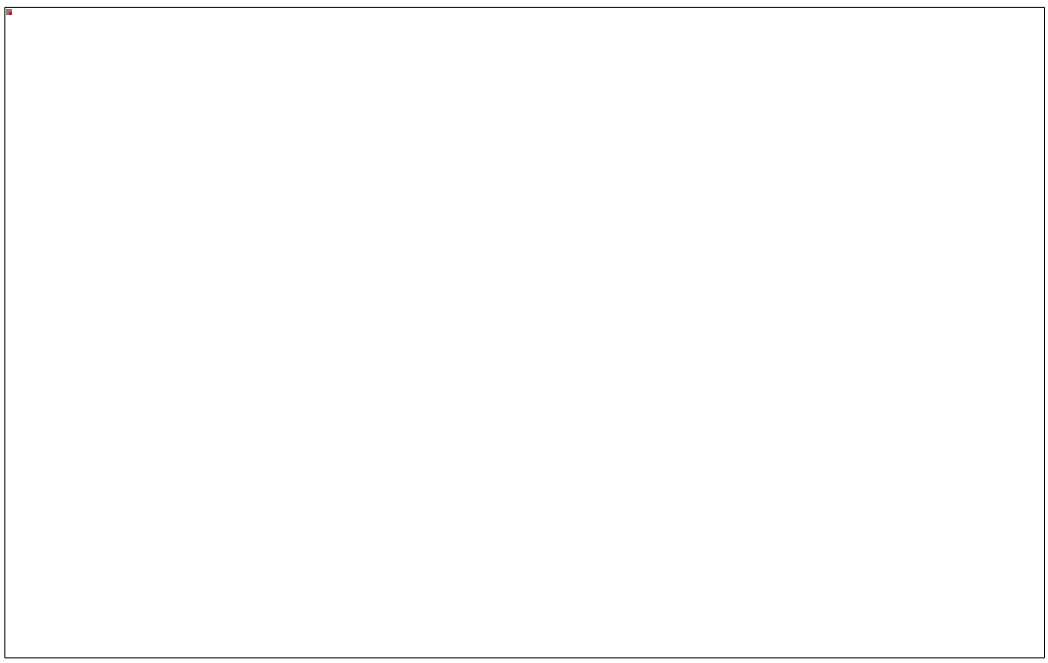
VIRSTOP can also be loaded high from either CONFIG.SYS or AUTOEXEC.BAT using normal DEVICEHIGH or LOADHIGH commands. Instructions for this can be found in chapter 2.5 of this manual. VIRSTOP also supports several command-line switches. See chapter 2.4 for details.

Since VIRSTOP is both a device driver and a normal executable file, it can also be started directly from the command line with the command

```
C:\> F-PROT\VIRSTOP
```

1.3 Basic Procedures

Menu items are selected by highlighting them with the arrow keys and pressing ENTER or by typing the highlighted letter in the command.



The Main Menu of F-PROT. Menu items are selected by highlighting them with the arrow keys and pressing ENTER or by typing the initial, highlighted letter.

The most common way of searching for viruses with F-PROT is the "Secure Scan" method. "Secure Scan" can be started by selecting **Scan**, **Method/Secure Scan**, followed by **Begin Scan**.

The menu selections are made with arrow keys and confirmed by pressing ENTER or canceled by pressing ESC. When **Yes/No** selections have to be made the space bar is used to toggle the choices.

The space bar is used for toggling different options in submenus. When the desired set of options is selected, it is accepted with ENTER. Any menu can be cancelled by pressing ESC.

In the virus description menu and other lists, selections can be made using the PageUp and PageDown keys in addition to arrow keys. A name can also be selected by typing the first few letters of it. If more information on the Ping Pong virus is needed, just type "p—i—n—g". F-PROT will display the information for that virus as soon as the string given identifies it unambiguously.

Secure Scan can be started directly from the command line with the command

C:\> **F-PROT /HARD**

The contents of a floppy disk can be examined and, if necessary, disinfected with the command

C:\> **F-PROT A: /DISINF**

More information on the different command line parameters can be obtained with the command

C:\> **F-PROT /?**

1.4 The Components of the F-PROT Package

- F-PROT.EXE — the main module of the F-PROT package with virus scanning and disinfecting features. F-PROT is able to scan hard disks, removable media, floppy disks, and network file servers.

- VIRSTOP.EXE — the virus-intercepting TSR-program that offers active protection from viruses. Normally, VIRSTOP takes up only about 20 KB of memory and, if the /DISK parameter is used, only about 3.5 KB. It is very compatible with different applications and local area networks. We recommend keeping it active at all times.
- F-TEST.COM — a small accessory program for verifying that VIRSTOP is operating. The usage of F-TEST is described in chapter 4.1 of this manual.
- F-CHECK.EXE — Integrity Checker
- SIGN.DEF — a database containing signatures for viruses and Trojan horse programs. F-PROT has at least two different signatures for each virus. This file is encrypted.
- USER.DEF — an optional file containing additional signatures defined by the user.
- SETUP.F2 — used for storing user preferences.
- ENGLISH.TX0, ITALIANO.TX0, SWEDISH.TX0, VIR-HELP.ENG, VIR-HELP.ITA... — used for supporting various languages.
- F-PROT.ICO and F-PROT.PIF — icon and program information files to be used with Microsoft Windows

1.5 Minimum Configuration

The F-PROT package takes up less than one megabyte of hard disk space. If even this is a problem, some of the files can be left out to conserve storage space. The files that are absolutely required for normal operation of F-PROT are

- F-PROT.EXE
- SIGN.DEF
- ENGLISH.TX0 or any of the TX0 language files

If you choose to use this minimum configuration, you won't be able to change the language used or view the virus descriptions. Otherwise the program will function normally.

To install VIRSTOP, only one file is needed: VIRSTOP.EXE. The program is fully self-sufficient, as the file also includes all signatures used by VIRSTOP.

2 Installation and Settings

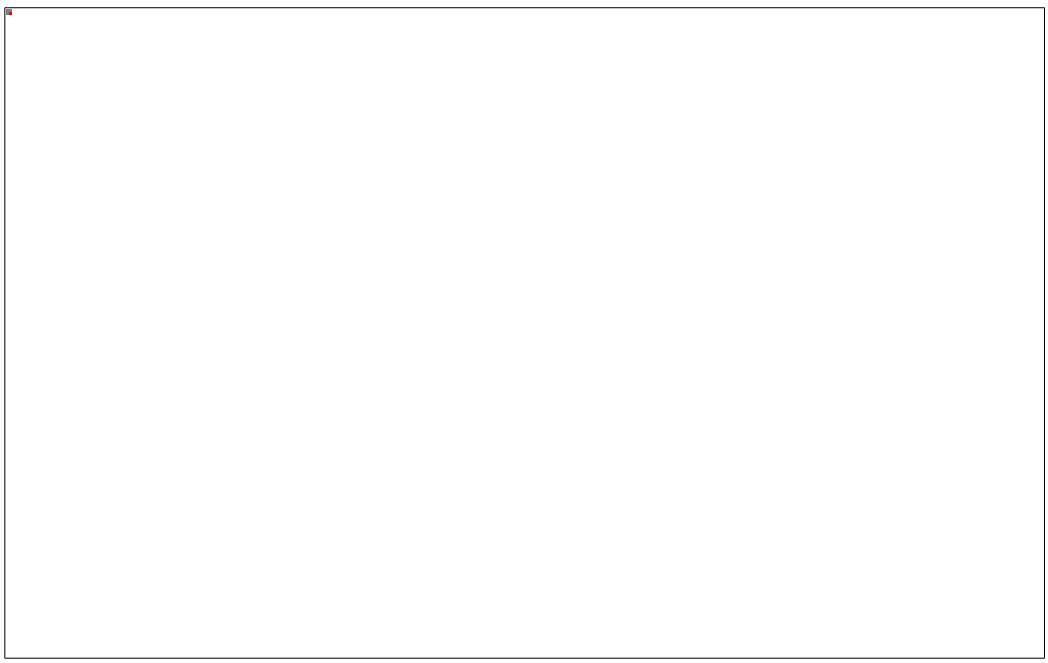
F-PROT can be used directly from the installation diskette. We recommend, however, installing F-PROT to the hard disk, since this speeds it up and makes it easier to use. Keep the original diskette in a safe place, as it is needed in case of a virus outbreak.

2.1 Installation on the Hard Disk

F-PROT is installed to the hard disk by running it from the installation diskette

```
C:\> A:
A:\> F-PROT
```

Before running F-PROT for the first time, make sure the diskette is write-protected. F-PROT starts by loading the virus descriptions and signatures to memory and then proceeds to check the RAM for active viruses. After completing these preliminary steps, F-PROT displays the main menu.



The main menu of F-PROT

The installation of F-PROT is fairly simple, and can be done either manually or automatically.

The automatic installation assumes that F-PROT is present on a diskette, and the intention is to install it on the hard disk. To install, run the program and select **Install** from the **Install** submenu.

The installation routine will ask for a directory for F-PROT. The recommended directory is C:\F-PROT. After all necessary files have been copied to the hard disk, a company specific message string can be entered (e.g. "Virus infection — call Peter at extension 1234."). This warning is displayed by VIRSTOP when an infected application is launched. The message can be changed later, if desired.

After the installation is completed, VIRSTOP can be activated by selecting **VIRSTOP** from the **Install** menu. This menu choice will enter the necessary commands into AUTOEXEC.BAT for the automatic loading of VIRSTOP at boot time. VIRSTOP is the memory resident part of the F-PROT-package. It should be used at all times.

VIRSTOP can be activated at once by restarting the computer. This can be done by pressing the keys Control-Alt-Del simultaneously.

Manual installation involves copying all the files in the package to a separate directory on a hard disk or on a network server. One line is then added to the AUTOEXEC.BAT file to load the VIRSTOP.EXE program. If you are unsure about how to create directories and use a text editor, please use the automatic installation.

Viruses can be searched for or removed and the settings can be changed after starting F-PROT with the command

C:\> **F-PROT\F-PROT**

2.2 Selecting the Language

The program is available in several different languages, with more to come. When F-PROT.EXE is run it searches for *.TX0 and VIR-HELP.* files, which contain all the necessary texts in each language. If your version has multiple language files, the **Language** option can be used to specify which language to use. This information will be stored in the SETUP.F2 file.

It should be noted that the answers for the "yes/no" prompts are not the same in different languages. For instance, in English the correct letters are Y/N and in Italian S/N.

If there's no configuration file (SETUP.F2) present, the default language depends on the country information returned by DOS — for example, if both English and Swedish language files are provided, but the current country is Norway, English would be used by default. You should select the correct language for F-PROT before installing VIRSTOP.EXE, so the correct version of VIRSTOP.EXE will be created.

After the correct language is selected once, F-PROT will remember it. If you find out that the program starts up using a language you don't know, you can restore the default settings by deleting the configuration file SETUP.F2.

2.3 Customizing the Message Displayed by VIRSTOP

The **Install** menu option is used for two different purposes. Besides copying the application to the hard disk, it is also used for building a language specific version of VIRSTOP. Please note that building a new version of VIRSTOP does not load it automatically. If you want to have VIRSTOP resident at all times, please follow the instructions in the next chapter.

2.4 Executing VIRSTOP at Startup

The **VIRSTOP** option in the **Install** menu will add the following line to the beginning of AUTOEXEC.BAT

```
C:\F-PROT\VIRSTOP
```

This command line will load VIRSTOP into memory every time the computer is restarted. The correct path to the F-PROT subdirectory is used, even if it is different from the default C:\F-PROT.

If VIRSTOP has already been installed this option can be used to uninstall it. In practice, this means deleting the command line from AUTOEXEC.BAT. After the next boot-up, VIRSTOP will not be active any more.

VIRSTOP can also be started from CONFIG.SYS by adding the following line

```
DEVICE=C:\F-PROT\VIRSTOP.EXE
```

When started from CONFIG.SYS, VIRSTOP is activated in an early phase of the boot-up sequence. *Note*: when HIMEM.SYS is used, VIRSTOP must be loaded after it. VIRSTOP can also be loaded high. Detailed instructions can be found in chapter 2.5.

In networked environments, F-PROT and VIRSTOP can also be installed on a file server. Generally, a single copy should be installed on a server, so that every single computer does not need to be updated when a new version of F-PROT is released. Detailed instructions for installing F-PROT on a network file server can be found in chapter 2.7.

2.5 Command-line switches supported by VIRSTOP

The main purpose of VIRSTOP is to stop the execution of infected program files. However, it also includes several other features. They are:

- automatic checking of the boot sectors on disks and diskettes
- the checking of program files when they are copied or when other operations are performed on them
- checking the boot sector of the diskette in drive A when the user presses Ctrl-Alt-Del

VIRSTOP supports the following command-line parameters:

/DISK

VIRSTOP stores the virus signatures on the disk, thus reducing the program's memory requirements. The /DISK parameter should not be used with the DEVICEHIGH= clause. DEVICE= and LOADHIGH, on the other hand, are completely functional. This parameter shouldn't be used when VIRSTOP is stored on removable media such as floppy disks.

/FREEZE

if the /FREEZE-parameter is used, VIRSTOP stops the computer when it finds a virus. VIRSTOP displays and names the virus and brings a customized message on the screen before stopping the computer. By using the /FREEZE-parameter, the risk of a virus succeeding to spread is minimized – the user cannot ignore a virus infection even if he wants to. This parameter should be used with care, since the user will not be able to save his work after warning, either.

/BOOT

this parameter commands VIRSTOP to check the boot sectors of all disks and diskettes when they are used. The check is made if any operation, like, for example, directory listing, is performed on the disk. The function can be turned off with the parameter /NOBOOT.

When somebody attempts to use an infected diskette, VIRSTOP brings the following message on the screen:

```
VIRSTOP alert! - virus on diskette
Press [ENTER] to continue.
```

/COPY

if the /COPY-parameter is used, VIRSTOP checks program files when they are read. This means that if programs are copied or edited, they are checked at the same time. This function can be turned off with the parameter /NOCOPY.

When a virus is found, the /COPY function announces it in the same way as when it notices an attempt to execute an infected file:

```
C:\TEST.EXE is infected with the Cascade virus.
```

And prevents the use of the program, whereupon DOS announces, depending of the version:

```
Cannot execute: TEST.EXE

Access denied: "test.exe"

Bad function
```

/WARM

If VIRSTOP is started with the /WARM parameter on, it checks whether the computer's drive A contains a diskette when the user presses Ctrl-Alt-Del. If the drive contains a diskette, its boot sector is checked for viruses. If the diskette is clean, the computer is rebooted, but if it contains a virus, the user is informed of the matter and advised to remove the diskette before booting the computer again:

```
VIRSTOP: Virus-checking A:
VIRSTOP alert! - virus on diskette
Press [ENTER] to continue.
Remove diskette and reboot again.
```

The use of /WARM parameter does not give a foolproof protection against boot sector viruses, because it cannot prevent booting from an infected diskette if the computer is reset by using the reset switch or by turning off power. In addition to this, some badly designed TSR programs hijack the Ctrl-Alt-Del - function for their private use, in which case VIRSTOP cannot perform the check. The check cannot be made under Windows, either.

When VIRSTOP is started without any parameters, it reports its status. For example:

```
C:\F-PROT\VIRSTOP

VIRSTOP is already installed.
VIRSTOP will scan programs when they are run.
VIRSTOP will scan programs when they are run or copied.
Diskette boot sectors are scanned when the diskette is accessed.
The A: boot sector is scanned when Ctrl-Alt-Del is pressed.
```

The check on executed files is always performed regardless of other parameters.

2.5 VIRSTOP for Windows

The VIRSTOP program functions also under Windows. Like its counterpart that functions under DOS, the Windows version is a memory-resident program that prevents viruses from being executed and infecting the computer.

Since the Windows support has been built in the internal structure of VIRSTOP, the program can be taken into use directly, without such additional steps as installing it in Windows. VIRSTOP notices when Windows is started, and automatically loads its Windows part into memory as one of the Windows device driver routines. The VIRSTOP.EXE program file contains the code for both Windows- and DOS parts of the program. VIRSTOP does not need any other files to function.

A Windows DOS window should not be used to load VIRSTOP, or any other TSR program, into memory. We recommend including VIRSTOP in the computer's AUTOEXEC.BAT file.

When an attempt to execute an infected program is made from Windows, VIRSTOP for Windows stops the execution and warns the user of an infected file. If VIRSTOP's /BOOT parameter is on, a corresponding warning is given of a diskette infected by a boot sector virus. The program's functioning can be tested by executing the familiar F-TEST.COM under Windows.

VIRSTOP for Windows has detected a boot sector virus on a diskette and prevented the diskette from being used.

VIRSTOP for Windows displays the virus warnings in text format to make sure that a user notices them. The warning box can be exited by pressing Enter, after which the previous program resumes its

execution. VIRSTOP for Windows monitors also the DOS windows opened from Windows.

If the keys Ctrl-Alt-Del are pressed, VIRSTOP for Windows cannot check the boot sector of a diskette in drive A even if the /WARM switch is turned on, because Windows interrupts the computer's functioning during a warm boot. The /BOOT switch, which causes VIRSTOP to check the boot sectors of all diskettes that are used in the computer, should therefore always be used.

The Windows routines of VIRSTOP have been made very small in order to save memory. The Windows support increases the memory requirement of VIRSTOP's DOS part only by 112 bytes. The device driver loaded during the Windows startup takes up approximately five kilobytes of Windows memory. The Windows support of VIRSTOP can, if necessary, also be switched off by using the command line parameter /NOWIN.

Important: VIRSTOP does not detect the same number of viruses than F-PROT's Secure Scan. The detection of some of the encrypted viruses needs so much memory and resources that it is not practical to search for them with a resident program. It is therefore recommended that VIRSTOP should only be used as one component of the virus protection. However, VIRSTOP is able to stop practically all of the common viruses.

2.5 Optimizing the Memory Usage of VIRSTOP

VIRSTOP can be loaded high. This will leave more base memory below 640 K for use by other programs. The use of upper memory varies between different versions of DOS. The instructions presented here apply to MS-DOS.

To enable loading VIRSTOP high, add the following lines to the beginning of CONFIG.SYS:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
```

To actually load VIRSTOP high, one of the following lines must be added in either CONFIG.SYS or AUTOEXEC.BAT after the above mentioned commands. In CONFIG.SYS:

```
DEVICEHIGH=C:\F-PROT\VIRSTOP.EXE
```

Or in AUTOEXEC.BAT:

```
LOADHIGH C:\F-PROT\VIRSTOP.EXE
```

If HIMEM.SYS is used, it must be loaded before VIRSTOP even if VIRSTOP is not loaded high.

If the PC is connected to a network, VIRSTOP should be run from the AUTOEXEC.BAT after the network software is installed. However, the correct load order varies between different types of networks, and you might need to test the loading order to find the optimum configuration.

VIRSTOP takes up approximately 20 kilobytes of memory. It is possible to reduce this amount by starting VIRSTOP with a /DISK command-line switch. This parameter allows VIRSTOP to swap signatures in from disk as necessary, which reduces the memory requirement of VIRSTOP down to 3 kilobytes.

If the amount of available memory is not critical, we recommend starting VIRSTOP without the /DISK switch, since it slows down the operation of VIRSTOP.

2.6 Other Settings

This option is intended for tailoring the program to specific tastes. Currently, it can be used to select whether the list of viruses should be displayed in line-by-line or column-by-column format. We recommend using the column-by-column format.

If any of the options are changed from their default values, F-PROT will ask if the changed settings should be saved when you exit from the program. If so, a file named SETUP.F2 will be created.

SETUP.F2 does not necessarily have to be in the same directory as the main program. Instructions on how to save SETUP.F2 into a different directory can be found in chapter 2.7, “Installing F-PROT Onto a File Server”.

2.7 Installing F-PROT onto a File Server

If the licence acquired by your company encompasses all workstations attached to the network, F-PROT can be installed onto the file server. This way, the program can easily be updated for all its users by simply replacing an older version in the file server with the latest one.

The command line in AUTOEXEC.BAT that starts VIRSTOP must be placed after the lines for the low-level network drivers. This will enable VIRSTOP to detect the read/write attempts to the network drive as well as to the local disks.

As network drivers may work in mysterious ways it is sometimes necessary to find the correct place for the loading of VIRSTOP by testing several alternatives. The first thing to do is to place the line that loads VIRSTOP as the last line of CONFIG.SYS.

After booting the computer it is possible to use the F-TEST application to test whether VIRSTOP is functional. F-TEST should be copied to a network drive and executed from there. If F-TEST reports that VIRSTOP is not functioning, the line for VIRSTOP should be moved to the AUTOEXEC.BAT after the first network drivers.

The same should be repeated until the line for VIRSTOP has been moved far enough for all the necessary network drivers to be loaded before VIRSTOP. At that time F-TEST will report that VIRSTOP is functioning normally.

The F-PROT directory in the file server can and should be write-protected. This will, however, make it impossible to save the default settings in the application directory. Each user needs to indicate a directory where the SETUP.F2 file should be saved. This is done by setting the environment variable FPSETUP to point to the desired directory.

We recommend adding the following command line to the AUTOEXEC.BAT in all workstations in the network.

```
SET FPSETUP=C:\SETTINGS\SETUP.F2
```

This makes it possible to save personal settings for each user on his local hard disk and still load F-PROT from the file server.

2.8 Registering the Program

Please return the registration card found in the program package to us as soon as possible. Only registered users are eligible for updates and technical support.

Send the registration card to your local dealer or F-PROT Professional Distributor or directly to

Data Fellows Ltd
F-PROT Support
Wavulinintie 10
FIN-00210 Helsinki
FINLAND

☎ +358-0-692 3622
fax +358-0-670 156

3 Scanning for Viruses

F-PROT finds practically all known viruses by using a method known as "scanning". This involves searching for a characteristic virus pattern or "signature" — a sequence of bytes which is very unlikely to be found anywhere but in a particular virus.

F-PROT uses at least two signatures for each virus. This means that F-PROT's hit rate is very high. A third signature is included for most viruses to speed up the Quick Scan function.

3.1 Quick Scan, Secure Scan and Heuristic Analysis

F-PROT can use three different methods to scan for viruses.

Secure Scan uses two different signatures for each virus. It will also search for the signatures in a large block of data — usually (but not always) located either at the beginning or the end of the file. This improves the chances of detecting any virus that has been created by modifying an older one — any change may cause a signature to be located at a different position within the virus, or it may even corrupt the signature itself, but the chances of a single change invalidating both of the signatures are practically zero.

Quick Scan is, as the name implies, a faster method than Secure Scan, but it is also less secure. It uses only one signature for each virus, and to speed things up further, Quick Scan does not spend time on an accurate identification of any virus it might find.

Quick Scan will just report a Jerusalem infection, while Secure Scan would report an infection by the Anarkia-2B variant of Jerusalem, for instance. Most users are not concerned with the accurate identification of any virus which might strike — all they want to know is whether or not they have a virus.

Quick Scan is almost as good at finding known virus variants as Secure Scan. However, Quick Scan is unable to find some encrypted viruses and some viruses written in high-level languages. If you select Quick Scan, you cannot select disinfection, as it requires an accurate identification. Because Quick Scan will not search for user-defined strings, the Targets option is disabled as well.

All methods that use signatures or known pieces of code to identify viruses are powerless before a new virus that has not been analysed and the signatures added to the software. The third search

method of F-PROT searches for viruses without prior knowledge of any particular virus. The search method is called *Heuristic Analysis*.

Heuristic Analysis uses a set of rules, instead of a signature database. Its purpose is to detect suspicious code. It is not foolproof and not recommended for the casual user. It will not detect all viruses and may sometimes produce false alarms, so it should be used with care. However, as it is not limited to existing viruses or variants of them, it is equally effective against new viruses.

Heuristic Analysis recognizes 95% of known viruses as suspicious; therefore it is likely that it will recognize close to 95% of all new viruses, too.

Heuristic Analysis is described in more detail in chapter 6.

3.2 Selecting the Scanning Options

The second option of the **Scan** menu, **Search**, is used to select the drives and directories F-PROT should search for viruses.

- **Hard Disk** — scans all partitions on the internal hard disk
- **Diskette Drive** — scans one of the 3.5" or 5.25" diskette drives
- **Network** — scans all the network drives
- **User Specified** — scans a drive, directory or file specified by the user. If a directory is selected F-PROT will scan all its subdirectories as well.

Some removable hard disks show up in the Diskette Drive menu. This is due the drivers used and is perfectly normal.

By using the **Network** option, the purity of a network server's hard disk can be checked from a workstation. It is sensible to open a separate user account for the purpose, one with a read authorization to all the files and directories on the network server, but with no write authorizations. When checking the purity of the network, it is recommended to log in with this user ID.

F-PROT can check the servers in most networks without any trouble, regardless of the operating system or the network type. The boot sectors of network servers can not, however, be checked, from a workstation, so they should be checked from the server itself.

The difference between selecting Diskette drive A: and selecting User-specified and entering A: is that in the former case F-PROT assumes you want to scan multiple diskettes, and after scanning each diskette, a report is given and you are prompted for the next diskette.

The search can easily be directed to a hard disk, a diskette drive, the file server, a freely selected directory and its sub-directories or to a single file.

The third option in the **Scan** menu, **Action**, is used to specify what action will be taken, should a virus be found. The default action is just to list the names of any infected files, but F-PROT can also disinfect most of known viruses.

If disinfection is needed, it can either be fully automatic, or F-PROT can prompt for confirmation before it attempts to disinfect any given file. Sometimes an infection cannot be removed. This happens if the virus has overwritten and destroyed the infected file.

Because it is almost always safer to delete infected programs than to disinfect them, F-PROT always offers deletion as an option — any infected file will first be overwritten several times and then deleted. It is possible to select automatic deletion or have F-PROT prompt for confirmation before it deletes a file.

Finally, an infected file can be renamed to avoid accidental execution if the file is kept for further study.

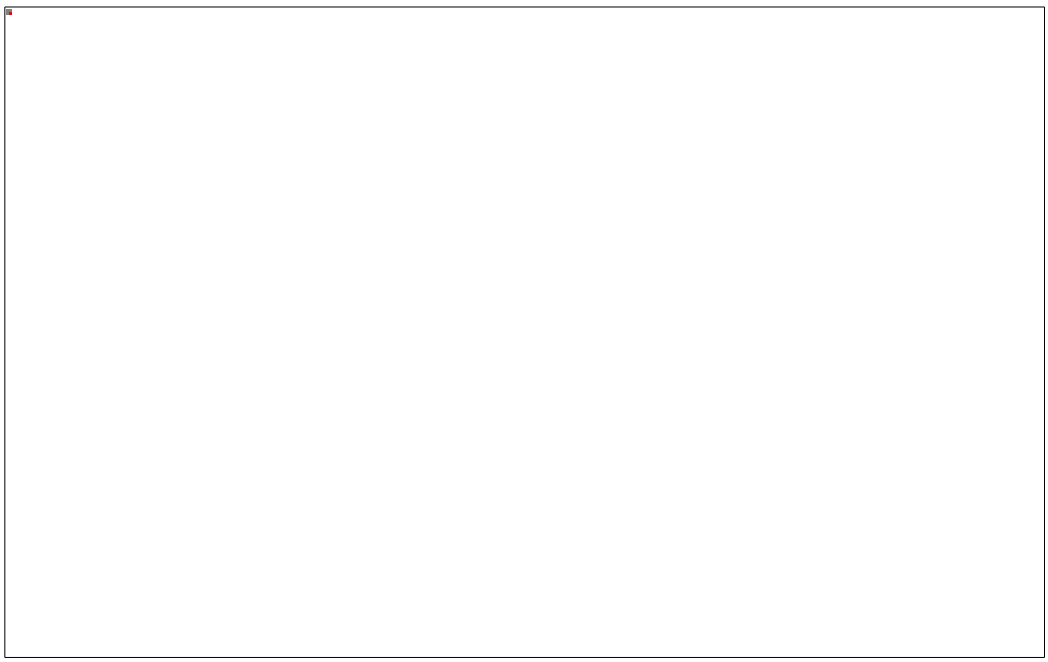
The alternatives for handling infected files are:

- **Report only** — the infected files are reported, but they are not disinfected
- **Automatic disinfection** — the virus is eradicated automatically
- **Disinfect/Query** — F-PROT prompts for confirmation before it attempts to disinfect

- **Automatic deletion** — any infected file will first be overwritten several times and then deleted
- **Delete/Query** — F-PROT prompts for confirmation before deleting the file
- **Rename** — the file name extensions of the infected files are changed from .COM to .VOM and from .EXE to .VXE to prevent unintentional execution.

The **Report only** -option can be recommended for most situations. If a virus is found in the computer, its variant name and version must be marked up together with the names of the infected files. This can be done by, for example, printing F-PROT's virus report on paper. This report is then used for finding out the source of the infection. All active copies of the virus must be found and destroyed, because otherwise it may appear again after some time.

The menu option **Scan/Targets** is used to select the types of viruses to search for. The default is to search for all kinds of ordinary viruses. It is also possible to search for boot sector viruses, file viruses, and Trojan Horses, in all different combinations.



In the Targets submenu, all selections are made with the space bar. When all selections have been made they can be approved by pressing the ENTER key.

Normally, all known viruses should be searched for, but in certain circumstances you may want to exclude boot sector viruses or file viruses.

For example, during the clean-up operation after an attack by a specific boot sector virus, file viruses need not be searched for on every single diskette.

F-PROT can be instructed to search for user-defined signatures. These are entered from the **New Signatures** option in the Viruses menu. The signatures are entered as hexadecimal strings. The signatures can also be entered directly to USER.DEF file with any text editor.

The setting **Packed files** in the **Targets** option in the **Scan** menu is used to determine whether F-PROT should search for viruses in packed files. F-PROT can find viruses in files packed with LZEXE, PKLITE, EXEPACK, DIET and ICE regardless of whether the infection has occurred before or after packing the file.

F-PROT can also find infections on files packed with other tools, if the infection has occured after packing. Unfortunately, it is not possible to check the "internal" purity of files packed with some other (less popular) tools. F-PROT will report these files and tell that it was only able to check them externally. This is generally not cause for alarm.

The **Scan** menu option **Files** is used to select what kinds of files F-PROT should scan. Most viruses will only infect normal executable files.

The default operation of F-PROT is to scan the most commonly infected filetypes: COM, EXE, OV?, APP, PGM and SYS. It is also possible to select "All files". This is not recommended in normal situations. It is advisable only if you are cleaning up after a virus attack — just to make sure the virus is not hiding in some obscure overlay file. It is also possible to add a set of file extensions to the list of files to scan. XTree Gold's XTR-overlays could be added to the group of scanned files, for instance.

3.3 Starting the Scan

When you have selected the desired options, you may start the scanning by selecting **Begin Scan** at the top of the **Scan** menu, either by moving the cursor there and hitting ENTER, or just by pressing B.

The small window at the bottom of the screenwill display the names of the files as they are scanned. The scanning can be aborted at any time simply by pressing the ESC key.

When the scanning is finished, a summary is displayed. If no viruses or suspicious programs were found, the summary is empty, but otherwise a detailed listing is produced when ENTER is pressed. This listing can be saved to a disk file or sent to the printer.

When a virus is found, it is important that the user can get additional information about it. This can be done using the **Viruses** menu.

In the **Viruses** menu and other lists, selections can be made using PageUp and PageDown keys in addition to arrow keys. A name can also be selected by typing the first few letters of it. For example, if more information on the Ping Pong virus is needed, just type "p—i—n—g". F-PROT will display the information for that virus as soon as the string given identifies it unambiguously.

In the virus list, selections can be made using PageUp and PageDown or arrow keys. A virus can also be selected by typing the first few letters of its name. The major name of each virus is displayed using yellow colour.

When a virus is found, it is important that the user can get additional information about it.

3.4 Disinfecting Files

F-PROT can be used to disinfect files. F-PROT does this by removing the virus code from the file and leaving the original code intact. Naturally, this can only be done if the virus has not harmed the original code in any way.

If the program code has been irreparably damaged by a virus, F-PROT will ask for permission to delete the file. The original program must then be reinstalled from backups. Generally it is preferable to reinstall the infected files. F-PROT can disinfect files reliably, but even so there are some situations in which it may be impossible to reach a satisfactory result.

When a file is disinfected, it is usually restored to its original state. There is one exception, however. In many cases, the disinfected program will have 1-15 additional meaningless bytes at the end. These bytes are added by many viruses to make the length of the program a multiple of 16 bytes. As the number of the extra bytes cannot be determined, they cannot be removed.

Normally, they will not have any effect, unless the program checks its current length. In that case, the program will report an incorrect length after disinfection, and must be restored from a backup.

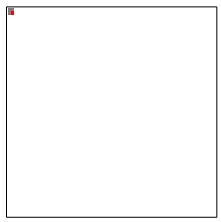
The F-CHECK program is usually able to fully restore files that F-PROT is unable to disinfect. However, in order to achieve this, the file in question must have been added to F-CHECK's database before infection.

F-PROT is capable of repairing infected files to working condition. To ensure problem-free operation, we recommend reinstalling the original file from a back-up.

In some situations, disinfection should be used instead of restoring originals. For instance, boot sector infections are easier to disinfect with F-PROT since the usual alternative is to make a back-up copy, format the disk, and restore the back-up.

This is especially important if the virus hides its code in the main boot record since main boot records cannot be cleaned with the FORMAT command.

3.5 Creating an Anti-Virus Diskette for Emergency Situations



To make it easier to recover from disk crashes, virus attacks or data loss due to user error, a tool diskette should be created. The diskette should include essential tools for disk repair, unformatting, undeletion, virus searching and disinfecting.

Because so many tools are available, an almost endless variety of emergency diskettes can be compiled. The simple diskette described below is meant to be used in case of a suspected virus attack. Different diskettes could be created to be used in other situations.

The diskette is first formatted as a system diskette with the command

```
C:\> FORMAT A:/S
```

and partition information of the hard disk is stored to the diskette with the commands

```
C:\> MIRROR /PARTN

C:\> UNFORMAT /PARTN /L /TEST > A:PARTINFO.TXT
```

Note that MIRROR and UNFORMAT are available only in MS-DOS 5. MS-DOS 6 does not include similar tools.

MIRROR creates a file PARTNSAV.FIL to the diskette. This file contains the partition info of the hard disk's logical DOS partitions and can be written back to the hard disk with the command UNFORMAT /PARTN. The PARTINFO.TXT file created by the UNFORMAT command contains information of all the partitions of the hard disks. This information cannot be recovered directly to the hard disk but it will be useful to an expert if the hard disk becomes corrupted.

Additionally, the following files are copied to the diskette:

- CHKDSK — for checking the integrity of disks
- BACKUP and RESTORE or similar programs — to make backups
- FDISK.EXE — for reconstructing the hard disk's main boot record
- SYS.COM — for reconstructing the DOS boot record
- UNFORMAT.EXE — to restore image data stored by MIRROR

and at least the following parts of F-PROT:

- F-PROT.EXE
- VIRSTOP.EXE
- SIGN.DEF
- ENGLISH.TX0
- VIR-HELP.ENG
- SETUP.F2

An AUTOEXEC.BAT file should also be created. An example of a suitable AUTOEXEC.BAT is shown below

```
@ECHO OFF
A:\VIRSTOP

ECHO This is the anti-virus disk of Sturdy Ltd.
ECHO If you have any problems, contact Fred
ECHO at extension 4456.
ECHO Press ENTER to execute F-PROT.
PAUSE > NUL
A:\F-PROT
```

If special device drivers are needed for the system's storage devices, they should be installed on the anti-virus diskette. This should be done with e.g. hard disks packed with Stacker or SuperStor, disks formatted with Disk Manager or similar products and some SCSI-drives. MS-DOS 6.0´s DoubleSpace compression system does not require a separate driver, as long as the diskette is formatted under MS-DOS 6.

It is also a good idea to write down the setup settings for the hard drives. These settings are stored in so-called CMOS memory and can be viewed using a setup program or the BIOS's built-in setup facility. Writing down the settings is important as some viruses zap the CMOS or the computer's battery can run down, wiping the settings from the CMOS. If the number of computers is large, all settings should be written down computer by computer and attached to the computer in question.

When the anti-virus diskette is formatted and all the necessary files are copied, we recommend making sure that a computer can be booted from it. If everything is in order, the main menu of F-PROT will pop up after the booting sequence.

In some cases, it may be a good idea to change the last line (A:\F-PROT) from the Autoexec.bat file of the diskette and run F-PROT in command line mode, as this is considerably faster than running the program in interactive mode from diskette.

We recommend setting **Secure Scan** and **Disinfect/Query** as defaults for F-PROT on the anti-virus diskette. These settings usually yield the best results in a crash recovery situation.

When the operation of the diskette has been verified and the defaults are set, the diskette should be immediately write-protected. It should be kept somewhere near the computer, so it will be readily available when needed.

4 VIRSTOP and F-TEST

The primary purpose of the VIRSTOP.EXE program is to prevent the execution of programs infected with known viruses. VIRSTOP installs itself in RAM as a standard TSR and intercepts the so-called "Load-and-execute" function (INT 21h). Because of this, whenever an attempt is made to run a program, VIRSTOP gets a chance to examine it first.

VIRSTOP uses a simple but fast search to check for viruses, but it does not make an accurate identification, Secure Scan is necessary for that purpose.

If VIRSTOP finds a virus, it will abort the execution of the program, display a message and return an error. For example, if you attempt to run a program infected with the Cascade virus with VIRSTOP active in memory, you might see something like this:

```
This program is infected with the Cascade virus.
Access Denied: C:\INF-PROG.COM
```

VIRSTOP has a secondary function as well - it checks for any active boot sector virus when it is run.

VIRSTOP includes one additional feature — it verifies its own integrity. It can detect whether it has been infected by a "stealth" virus — an ability which is rather unusual. This check is performed in a number of different ways, both by using the operating system's services and by bypassing them. VIRSTOP can also often detect attempts to run programs infected by a stealth virus, even if the virus is active in memory.

VIRSTOP takes up approximately 20 kilobytes of memory. It is possible to reduce this amount by starting VIRSTOP with a /DISK command line switch. This parameter allows VIRSTOP to swap signatures in from disk as necessary, which will cut the memory requirement for VIRSTOP down to about 3 kilobytes.

We recommend using VIRSTOP without the /DISK switch unless memory is at premium. This will ensure the best performance.

If VIRSTOP gives a virus warning, a system administrator should be called. An anti-virus tool disk like the one described in the previous chapter can be used to identify the virus accurately and to disinfect the system.

A more detailed description of the actions that should be taken when a virus is found is given in chapter 9.

It must be noted that it may not be possible to install VIRSTOP on machines equipped with a Cyrix processor, since some versions of these processors are not fully compatible with the Intel family.

4.1 Verifying the Functionality of VIRSTOP

F-TEST is used for verifying that VIRSTOP is installed properly and that it is functioning normally. F-TEST contains a signature that will be recognized by VIRSTOP. If VIRSTOP is functional, an attempt to execute F-TEST will result in the following message:

```
VIRSTOP is installed and active.
```

If F-TEST is run from Windows, DOS Shell or some other shell program, the message may read

```
Cannot execute C:\F-PROT\F-TEST.COM
```

This also means that VIRSTOP is running and will stop viruses from infecting the system.

If VIRSTOP is not loaded or it has been damaged, F-TEST will give the following warning

```
VIRSTOP is not installed or not working!
```

If you encounter this message, reinstall VIRSTOP immediately. Instructions for this are given in chapter 2.1.

5 Compatibility, Updates and Support

F-PROT is kept continuously up-to-date, since new viruses, new virus types and new versions of old viruses are constantly being created. Updates are also needed to support new versions of operating systems, LANs and other products.

5.1 Updates

The purchase price of the program license includes free updates for one year. Updates are sent to the address given in the registration card returned to the dealer.

New updates are released every 1-3 months. A virus newsletter is sent with every update. The newsletter includes current virus information with emphasis on events pertaining to Scandinavia. All new viruses found in Scandinavia are explained in more detail.

In case of a sudden virus epidemic, virus signatures can be added to F-PROT's USER.DEF file by the user. The signatures will be distributed as a news bulletin to each F-PROT user. Additional information about signatures will be available through your dealer.

5.2 Support

F-PROT program licence includes limited phone support from your dealer or directly from Data Fellows Ltd. The necessary contact information can be found from the end of the next chapter.

To receive the best possible help certain information should be made available when calling (where applicable):

- the version of the operating system in use (DOS, Windows)
- the computer architecture: the brand and model, amount and use of memory, BIOS version, display type
- LAN type, type of network adapter card, version of network operating system
- types and versions of drivers in use (mouse driver, SCSI driver etc.)
- F-PROT version number
- a description of the problem: a report given by F-PROT, error message given by the operating system, the procedure to generate the same situation

- a listing of the contents of the AUTOEXEC.BAT and CONFIG.SYS files or an MSD report from the machine

If your company has an F-PROT licence for several users, we recommend channeling the support calls via the support person or the EDP department. The user calling to get support may be asked to fax the previously listed information. This will often speed up the solving of the problem.

5.3 Common Questions and Answers

Question: The text "VIRSTOP error" appears when programs are executed. Why?

Answer: You have started VIRSTOP with the /DISK parameter and afterwards destroyed VIRSTOP.EXE from the disk. Alternatively, you have lost network connection with the server containing VIRSTOP.

If VIRSTOP.EXE is updated while the program is active **with the /DISK parameter on**, the end result is indefinite: the program may print rubbish on the screen or crash, because its virus descriptions are updated "on the run".

Question: The AS/400 connections seem to cause problem when VIRSTOP is active in memory.

Answer: Some versions of the AS/400 drivers behave in a non-standard way. In order to make the machine work correctly, VIRSTOP must be manually moved to the CONFIG.SYS. The correct order of loading is:

1. The possible DEVICE=DXMAMODs or other drivers which make electronic mail connections possible.

2. DEVICE=VIRSTOP.EXE (DEVICEHIGH and /DISK -parameters should not be used at the same time)

3. DEVICE=ECYDDX.SYS, DEVICE=EIMPCS.SYS

Question: Why does F-PROT, when the "VIRSTOP" function is used, announce "F-PROT cannot yet install VIRSTOP in a network"?

Answer: When F-PROT notices that a computer is connected to a network, it does not add the VIRSTOP-line to the AUTOEXEC.BAT, for it would probably not be located correctly in respect to the network drivers. In order for VIRSTOP to be correctly installed, the installation is left for the user.

Question: Why did installing VIRSTOP in a Novell network cause printing disturbances in the Windows environment?

Answer: The problem is rooted on the shaky support that DOS and Windows give for TSR sofware, and it may have its cause in any TSR program that takes into use the interrupt 21h.

This problem can be fixed by changing the printing device definition from LPT1 to LPT.OS2 or LPT.DOS in the Windows Control Panel.

All the LPT.XXX ports point to the same physical port. The difference with a normal LPT port is that Windows manages printing via BIOS. Originally, the LPTx.DOS and LPTx.OS2 ports were designed to guarantee compatibility between different user interfaces.

If this does not help, check the following:

Are the NETX.COM and EMSNETX or XMSNETX.COM that came with the Windows 3.1 in use instead of the ones that came with the DOS 5.0?

Change the Timeout value in the Control Panel "Printers" -dialog into lesser or greater one.

Is the "Fast Printing Direct to Port" in the Control Panel's Printers/Connect -dialog covered with an "X"? If so, check whether removing it affects printing.

Check the value that is given to your printer driver in the [PrinterPorts] part of the WIN.INI. The last number should be at least 90.

Is it absolutely sure that the workstations have enough free disk space and that the TEMP environment variable is correctly defined?

Is the FILES= definition in the SHELL.CFG and in the workstations' CONFIG.SYS great enough? (60)

If PRINTCON is in use, check its definitions. They should be:

Suppress Form Feed	*Yes*
File Contents	*Byte Stream*
Print Banner	*No*
Auto Endcap	*No*
Enable Timeout	*No*

These parameters can also be set with the command

CAPTURE NB NA TI=0 NFF NT

5.4 Further Development of F-PROT

The F-PROT anti-virus program is under constant development.

Any suggestions for improvements are appreciated. If you have suggestions for new features, please contact Data Fellows directly at:

Data Fellows Oy
F-PROT Support

Wavulinintie 10
FIN-00210 HelsinkiFINLAND

☎ +358-0-692 3622
fax +358-0-670 156
Internet e-mail: F-PROT@DF.elma.fi
X.400: S=F-PROT, OU1=DF, O=elma, P=inet, A=mailnet, C=fi.

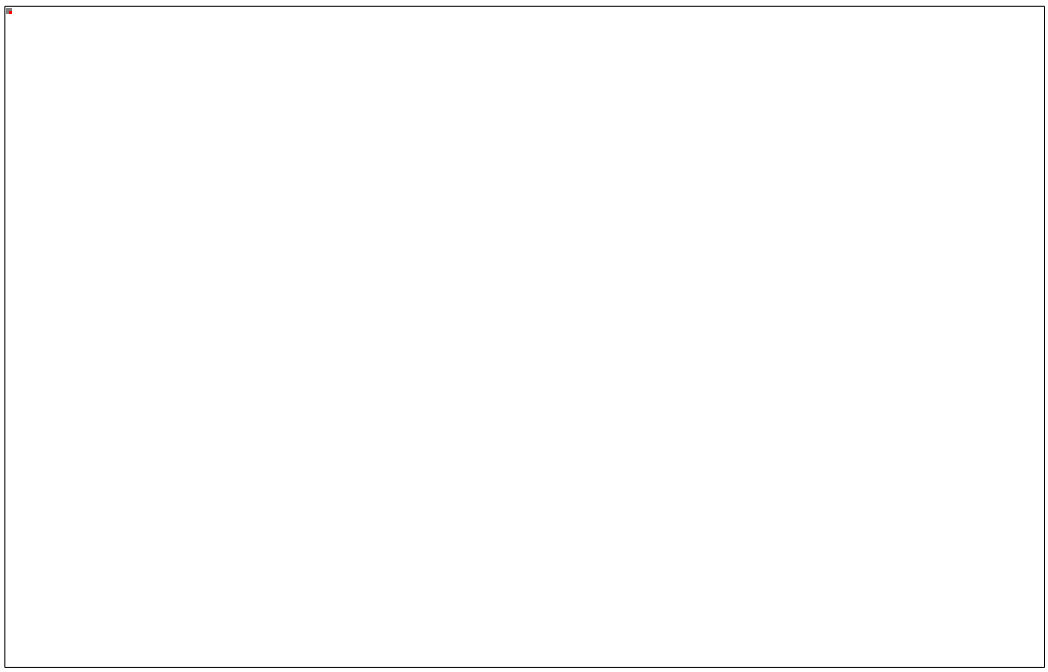
Table of Contents

Advanced features

6 Heuristic Analysis

Signature-based virus scanning is not the ultimate solution to the virus problem. When an up-to-date scanner is used, one can be fairly certain that all known viruses will be detected. The scanners may or may not detect new variants which have been created by modifying older viruses, but if a new virus is written entirely from scratch, it will probably not be detected by using any existing virus signature.

A virus may be detected by a generic monitoring program when it activates — perhaps when it tries to perform some suspicious action, such as reformatting the hard disk. It may also be detected by a checksum program, which detects changes to files or boot sectors after they have been infected. Nevertheless, it is preferable to try to detect the presence of the virus before running a copy of it.



The Heuristic Analysis can be started by selecting Heuristic Analysis as the search method and Starting Search.

Heuristic (rule-based) analysis is a new technology which, as the name implies, attempts to analyse programs and reports any suspicious code. This approach is not flawless — some viruses cannot yet be detected in this way and an occasional false alarm can be expected.

Heuristic analysis is, however, a promising way of detecting new virus strains and the technology is under constant development. Currently,

the heuristic algorithms used by F-PROT can detect up to 95% of all viruses. This gives a very high degree of protection against new viruses.

It must be noted, however, that a great number of normal DOS programs also contain unusual features. Many programs, for example, take advantage of the undocumented features of the operating system and use the hard disk directly, bypassing the normal services. Because of this, the Heuristic Analysis produces a certain number of false alarms. However, the Analysis gives false warnings of very few programs.

Please note that it is not suggested to run Heuristic analysis on all files. The analysis treats all scanned files as executable programs and will complain about normal data files as they do not contain valid executable code.

Heuristic analysis classifies suspicious code into four categories:

- certain infections
- probable infections
- suspicious code
- unusual code

Normally, only certain and almost certain infections are reported.

When a heuristic search is initiated, F-PROT begins to check through files using Secure Scan using it's normal search signatures and algorithmic methods. If a file is found to *not* contain a virus, it is searched also with heuristic methods. When F-PROT found a suspicious file, it states either:

```
C:\TEST.EXE seems to be infected with an unknown virus.
Please contact Data Fellows Ltd. or FSI and send us a copy for analysis.
```

or:

```
C:\TEST.EXE contains virus-like code.
Please contact Data Fellows Ltd. or FSI to check if this is a known
false alarm or send us a copy for analysis.
```

The Heuristic Analysis reports should be taken seriously. A file must meet several different conditions before it is suspected of having been infected. In the tests that Data Fellows performed using the new Heuristic Analysis, not a single false alarm was produced.

By starting F-PROT with the command line parameter /GURU, it is possible to make F-PROT also report suspicious and unusual code. This is done with the command

```
C:\> F-PROT /GURU
```

By using the /GURU -option, several different messages may be produced when suspicious code is found in a program, some of

which are nearly certain to indicate a virus infection. Here are some examples:

> This program contains several features which are normally only found in virus programs. It is almost certainly virus-infected.
>
> This program contains a virus which stays resident in memory when an infected program is run.
>
> This program contains a primitive virus,which is located at the beginning of the file.

Other messages might indicate a virus infection, but occasionally they are just false alarms. The less serious messages include:

> This program moves itself to a different area of memory using a method which is normally only used by viruses.
>
> This is a self-modifying program, which may indicate a self-encrypting virus or just unusual code.

Finally, there are a few messages which do not indicate a virus infection, only that something unusual has been found, such as:

> This file is packed using PKLITE, LZEXE or a similar program. It may have been infected before it was packed, but this program is not yet able to determine if this is the case.
>
> Some code has been added to the end of this file, but it does not appear to be a virus.

A false positive can be expected occasionally. However. if you encounter several non-related files giving similar warnings it is very probably caused by an unknown virus. If you have any questions, please contact F-PROT Support.

7 Command Line Mode

F-PROT.EXE is usually run without any parameters and will then enter the interactive mode. It is also possible to run F-PROT in command line mode, which makes it easy to tailor F-PROT to perform different kinds of searches by using different batch files.

F-PROT can be executed every time the computer is rebooted by adding the proper command line to AUTOEXEC.BAT.

If F-PROT is used from a diskette, it may seem slow because it reads the text information it needs from the diskette dynamically. If the program is run in the command line mode, it will work almost as fast as if run from the hard disk.

F-PROT can be used from the command line as follows:

F-PROT [drive, file or directory] [parameters]

If a drive, file or directory is given, F-PROT will enter command-line mode, unless the /INTER parameter is given as well. The command-line parameters are

```
/640        Only scan 640K of memory.
/ALL        Check all files.
/ANALYSE    Use heuristic analysis, instead of signatures.
/APPEND     Used with /REPORT - append to existing report.
/AUTO       Automatic deletion/disinfection.
/BEEP       Sound an alarm if a virus is found.
/COMMAND    Force command line mode.
/DELETE     Delete all infected files.
/DISINF     Disinfect whenever possible.
/EXT=       Specify default file extensions to scan.
/GURU       Report with more detail when using heuristic analysis.
/HARD       Scan all DOS partitions on the hard disk.
/HELP       Display this list.
/INTER      Force interactive mode.
/LIST       List all files checked.
/MONO       Use monochrome mode on color displays.
/MULTI      Scan multiple diskettes.
/NET        Scan any network drives found.
/NOBREAK    Do not abort scan if ESC is pressed.
/NOMEM      Skip initial memory scan.
```

```
/NOSUB      Do not scan subdirectories.
/NOWRAP     Do not wrap text in reports.
/[NO]BOOT   [Do not] scan boot sectors.
/[NO]FILE   [Do not] scan files.
/[NO]PACKED [Do not] scan inside packed files.
/[NO]USER   [Do not] scan for user-defined patterns
/PAGE       Pause after each page (command-line mode)
/QUICK      Faster search, but not as accurate.
/RENAME     Rename infected files to *.VOM or *.VXE
/REPORT=    Send the output to a file.
/SILENT     Don't generate any screen output.
/VERSION    Return with version number as an ERRORLEVEL value
```

Some examples of useful command lines are:

- scan a diskette using the Secure Scan method

  C:\> **F-PROT A:**

- scan multiple diskettes with automatic disinfection

  C:\> **F-PROT A: /MULTI /AUTO /DISINF**

- make a secure scan on drive C: and print a list of scanned files to LIST.TXT

  C:\> **F-PROT C: /LIST /REPORT=LIST.TXT**

- scan all files on all the partitions of the hard disk

  C:\> **F-PROT /HARD /ALL**

- scan all files on drive D: and E:

  C:\> **F-PROT D: E: /ALL**

- scan all network drives

  C:\> **F-PROT /NET**

- scan the public domain directory on the file server

  C:\> **F-PROT I:\PD**

- scan just the root directory of drive C:. It is a good idea to include a similar line to the end AUTOEXEC.BAT, so F-PROT will also make sure that there are no viruses active in memory

  C:\> **F-PROT C:\ /NOSUB**

You can interrupt a scan by pressing ESC, unless the /NOBREAK switch is used.

When F-PROT is run in command line mode, it will return an exit code. It can be checked with the DOS ERRORLEVEL command.

0 Normal exit — nothing found

1 Abnormal termination — unrecoverable error (usually a missing system file)

2 Self test failed — program has been modified

3 A Boot/File virus infection found

4 Virus signatures found in memory

5 Program terminated with Control-C, Break or ESC

6 At least one virus was removed

7 Out of memory

8 Using F-PROT under Windows and OS/2

F-PROT can be used with Windows 3.X, OS/2 1.X and OS/2 2.0 as a normal character-based application. It can be executed in its own window or as a full-screen application.

A mouse cannot be used with F-PROT, however. When using F-PROT with Windows or OS/2, an icon for launching F-PROT can be installed in the Program Manager or Workplace Shell. A suitable icon is supplied on the F-PROT disk in the F-PROT.ICO file. When operating under Windows, the file F-PROT.PIF should also be copied to the F-PROT directory from the installation disk.

VIRSTOP is also compatible with Windows. If the execution of an infected program is attempted from Windows, VIRSTOP will deny access to the file.

Under OS/2, VIRSTOP functions normally. We recommend, however, that VIRSTOP is loaded from AUTOEXEC.BAT instead of CONFIG.SYS.

OS/2-based systems can use different file systems. When a FAT-based file system is used, both F-PROT and VIRSTOP function normally in the OS/2 DOS box.

If the High Performance File System (HPFS) is used, programs that are named using the FAT convention (8 + 3 characters) are checked normally. The boot sector, however, cannot be examined, as its format is different from FAT disks. Boot sectors on FAT hard disks and diskettes can be scanned normally.

If you wish to be able to check all files on a HPFS disk, ask about F-PROT's OS/2 version from your local dealer.

9 Automating Scans under DOS with the F-AUTO

A small utility program called F-AUTO.EXE has been included on the F-PROT update diskettes. F-AUTO executes a freely program that can be freely selected on user-determined intervals. By using F-AUTO, it is possible to perform the F-PROT check automatically on , say, every third day. To accomplish this the following line is to be added to the workstation's AUTOEXEC.BAT:

```
F-AUTO 3 F-PROT /HARD
```

The above example assumes that F-AUTO and F-PROT are in the path. Usually it is a good idea to execute a batch file instead of running F-PROT directly, so the batch can check the return values returned by F-PROT. Here's on example on how to execute a batch file every three days:

```
F-AUTO /h 3 %COMSPEC% /E:1024 /C C:\F-PROT\FP.BAT
```

The parameter /h prevents F-AUTO from printing anything on the screen, 3 is here the desired interval in days. The last parameter on the line is the program that will be executed.

Because, in this example, the batch file FP.BAT will be executed, it must be started by using the DOS command interpreter. %COMSPEC% starts the command interpreter regardless of its position on the disk or whether COMMAND.COM, 4DOS.COM or NDOS.COM is in use. The commands /E:1024 and /C are needed in order to execute the batch file normally.

If the day parameter given to F-AUTO is 0, it will execute the desired program when it itself is executed for the first time in a day.

Note: If you're running a shared copy of F-AUTO through a network, set the environment variable F-AUTOPATH to point to a local directory. Otherwise all users will share the same F-AUTO.CFG file, which F-AUTO uses to keep track of the execution dates.

Computer Viruses

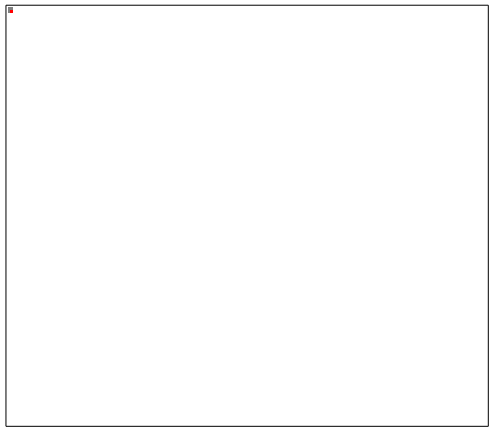


Table of Contents

10 What To Do When a Virus Strikes

The longer a virus infection remains undetected, the more dangerous and difficult to eradicate it becomes. It is possible to detect an infection by observing certain aspects of the operating and file systems.

Common signs of an infection are, for instance, a general slowing down of the whole computer, a slight delay every time an application is executed, abnormal read and write operations and unexplainable changes in executable and other files.

The following aspects should be taken into account when trying to assess the possibility of an infection:

- increased use of memory — a virus is active in memory
- increased use of disk space — the virus has attached itself into many files
- growth of files — the virus code increases the size of executable files
- a change in the latest alteration date of files, apparently without a reason — the virus has altered the program and neglegted to hide the change
- abnormal write-protected errors — a badly written virus is trying to infect a write-protected diskette. The virus does not check if the diskette is write-protected and generates an error from the operating system.
- The sudden appearance of hidden files on a disk or activities on normally unused areas of the disk — a virus is trying to hide its existence by storing its code in places rarely checked.

Unfortunately, the symptoms of a virus infection are usually so difficult to notice that the normal end-user does not pay any attention to the first visible signs of an infection. If the memory resident VIRSTOP is active in memory it will usually give the first warning of something being wrong.

An active virus has been found from memory. The advice F-PROT displays should be read carefully and acted upon.

If a virus has passed into the active phase of its existence, it is usually very easy to detect:

- files disappear
- the hard disk is formatted
- the computer does not boot
- information changes
- certain files cannot be loaded or executed
- the virus gives some other visible signs like writing messages to the screen or bouncing a ball from corner to corner

Some examples of different virus actions can be found from F-PROT's Virus Descriptions database.

If your computer is infected with a virus — *Don't panic*! Sometimes a badly thought out attempt to remove a virus will do much more damage than the virus could have done. If you are not sure what to do, turn your computer off and contact an expert.

The first actions after discovering a virus infection should be:

- Switch off the power
- Inform the system administrator and all those people to whom you have given information on a diskette or delivered executable files
- Put a note on the infected computer so that it will not be used before it has been disinfected

- Disinfect the infected files or reinstall all applications. Formatting the hard drive is almost always unncesessary. If the infection is a boot sector infection, simple DOS Format may not even eradicate the virus.

Finally, keep in mind that some viruses may interfere with the disinfection operation if they are active in memory at that time. Before attempting to disinfect you must boot the computer from a clean system diskette.

When booting the computer from a diskette, it must be noted that a mere Ctrl-Alt-Del is not, in some cases, enough to remove a virus from memory. A momentary disconnection of power from the computer ensures the purity of main memory.

It is often a good idea to prepare an anti-virus diskette which contains the tools needed to analyse and clean an infection. The preparation of an anti-virus diskette is explained in chapter 3.5.

If the infection is widely spread, it may be wise to make a special edition of the anti-virus diskette. In addition to copying F-PROT on the diskette the following line should be added to the AUTOEXEC.BAT.

```
A:\F-PROT /HARD /DISINF /AUTO /REPORT=C:\FP.REP
```

When an infected machine is booted from the specially prepared diskette, all infected files are automatically disinfected, and a report is stored in the root directory of drive C.

After all infected computers have been disinfected, all data and program diskettes should be checked. Diskettes containing only documents do not have to be checked if the virus in question is not a boot sector virus, but it might be a good idea to check everything — there might be other viruses in circulation at the same time.

All files should be backed up regularly. Current back-up copies not only help when recovering from a virus infection, but can also save the day after hard disk crashes and accidental erasure. Remember: an anti-virus program is not a substitute for frequent back-ups.

11 Computer Viruses: Background Information

Information security is a major problem in personal computers, because the internal protection mechanisms are inadequate. In current microcomputers, it is practically impossible to monitor the execution of a program once it is launched. Programs can do almost anything, including the destruction or alteration of data, without asking for permission or warning the user. In the most common microcomputer operating systems, there is no provision for different user and program access privileges. Because of this, anybody using a given computer can perform any operation.

The inherent problems are compounded by the fact that most users know little or nothing about what goes on in the computer when they start their favourite word processing or spreadsheet program. Software has become so complex that even experts have no means of verifying all the functions of a given program.

All sufficiently complex software contains errors in its design: the more complex and large the program, the more "bugs" are present. These bugs may, for example, be flaws in the basic design or strange things happening only in unlikely situations that have been overlooked during the programming.

When a user executes a program in a computer, he effectively gives the program the same rights and access privileges he has himself. These rights include the ability to modify or even destroy data permanently. If a program goes haywire and decides to wipe out all the poor soul's documents, there is usually nobody to stop it.

The frightening thing about these problems is that they may not be accidental. A programmer has full access to the resources in his clients' computers whenever his program is used. If he wants to cause damage, he can hide destructive behaviour into his program and, for example, design a time bomb of sorts to be triggered only after years of using the program. When the fatal day arrives, there is usually no way to find out who is responsible for the damage.

It follows that everybody simply has to trust the program manufacturers ability to control their programmers to play by the rules.

11.1 Viruses, Trojan Horse Programs, and Other Malicious Software

The oldest form of computer sabotage is to develop programs that have innocent names or even seem to do something interesting or useful when started, but contain traps that eventually do something undocumented. This is usually something undesirable, like wiping out files or whole disks. This kind of malicious program is called a *Trojan horse*, since it seems harmless on the outside and its documentation does not mention its "side-effects".

A Trojan horse may be effective in destroying data within a single system, but its efficiency depends on people copying and installing it into other computers. If its malignant behaviour is noticed, the program is not likely to be copied any more and will in some sense "die". Therefore, programs capable of copying and installing themselves are far more effective in spreading from computer to computer and site to site. A program of this kind is usually called a virus.

A very simple definition of a computer virus is: "A program that modifies other programs by placing a copy of itself inside them."

This definition is somewhat simplified, but it is sufficient to demonstrate the major difference between viruses and Trojan horse programs, which is that the viruses replicate while the Trojan horses do not. However, this simplified definition of a virus is not wholly accurate for some of the virus types, such as the so-called "companion" viruses or viruses that modify the partition information to get their own code executed.

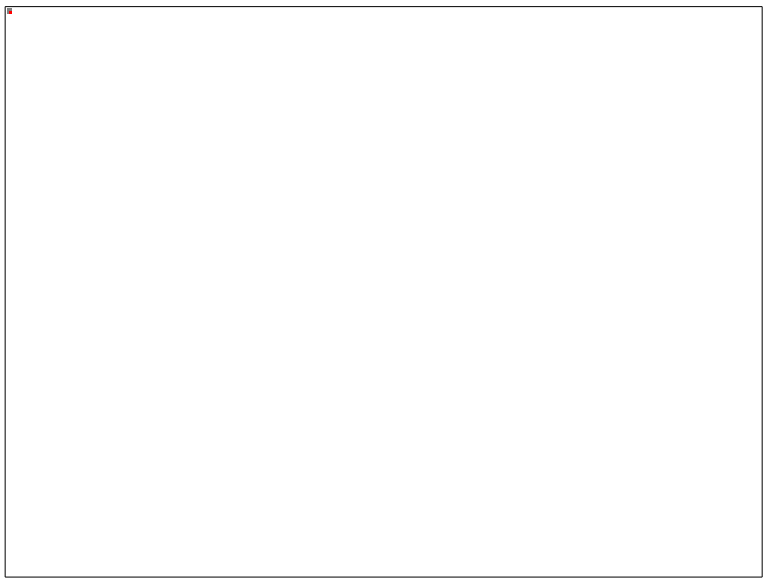
A virus "lives" as a parasite of a host program. It is usually very small and can therefore go unnoticed and infect systems without any visible symptoms. It can be constantly active, which means that it is capable of copying itself to other host programs, usually residing on computer disks and diskettes. When an infected diskette is used in another system or executable code is transferred by other means, the virus has a chance to conquer one more computer and start spreading even further. Viruses are quite independent and can infect whole companies very quickly without being noticed.

A virus usually has a latency period, during which it will do nothing but copy itself. If the virus is designed maliciously, it may then trigger and cause losses to the user. The nature of the trigger is entirely up to its programmer and the world has indeed seen some truly innovative examples of different triggers in the past few years. Some viruses trigger on April Fool's Day, some on Friday the 13th, some trigger at constant intervals. Some viruses cause mistakes in spelling to appear on printed documents that look perfectly all right on screen. Some viruses trigger when certain politicians' names are typed on the keyboard and attach insulting

comments to the names in real-time. Some viruses will create havoc only after they have installed a copy of themselves in a certain number of disks or after a certain period of time from the first infection, thereby making sure that everybody else gets a chance of getting infected before the virus is noticed for the first time.

The most common carrier of a virus is an infected diskette that is transferred from one computer to another. The viruses are contained either in programs on such a diskette, in the operating system itself or even in the boot sector of a perfectly innocent-looking data diskette (so called boot sector viruses). *Even a seemingly empty, formatted diskette can infect a system, if the computer is booted with the diskette in the drive.*

Diskettes are often infected by computers used by many people, such as print servers.



Diskettes are the most common source of infection in personal computers. Viruses can also spread via LANs and modem connections if programs are transferred. An infected file server is an especially virulent carrier that can quickly infect all the workstations attached to it.

Local area networks that use file servers or shared disks can also spread viruses, if executable program code is copied over the network or applications are run from the server. If all network traffic consists of pure data, viruses cannot spread. The same applies to modem connections. Fortunately, programs on file servers are usually located in protected directories and therefore cannot be infected.

The most risk-prone programs are illegally copied applications and games, because their origins are usually very varied. An illegally

copied program may travel a long way before reaching its eventual user and be infected many times on the way.

The first DOS viruses were found in 1986. They have found their way to all countries where personal computers are used, including Scandinavia. The number of known viruses increases steadily. Right now the number of known viruses is somewhere between 3000 and 4000, and it doubles annually. This means more than 4 new viruses each day.

It seems that destructive programs are more of an ethical and sociological problem than a technical one. Trojan horses and viruses are designed and released by people, who must have a motive for their malicious work. As their programs are very counter-productive, it would seem that no such motive exists; however, the overhelming number of viruses in the wild nowadays makes it obvious that there must be many reasons good enough to design a malicious program. Hopefully the situation will get better as the magical glow surrounding evil programs fades away.

11.2 Situation in Scandinavia

Scandinavia is often used as a test market for new hardware or software products due to the high level of computerization and the habit of the Scandinavians to always buy the latest, fastest, best and most expensive.

We use Scandinavia here as an example area. The history of viruses in Scandinavia resembles the situation in other areas. The ration between different viruses found in Scandinavia and the number of computers or people is extremely high, however.

Scandinavia has, indeed, not been spared from the damage caused by computer viruses. In Finland alone, over 90 different viruses have been found. In the whole of Scandinavia several hundred viruses have been reported. Thousands of companies have suffered from virus attacks.

The first virus spotted in Scandinavia was “Brain”, in 1988. After that the number of virus observations grew steadily, but in the beginning of the 1990s a sudden virus boom began. Currently, new viruses are found weekly. New viruses arrive in Scandinavia quite quickly, typically within a few weeks after the first reports from other countries.

It seems that Scandinavia acts as a gateway between the east and the west. A considerable number of viruses are developed in the former Soviet Union and in Bulgaria. These viruses arrive to Scandinavia very quickly. Typically viruses arrive via floppies received from partners or principals, through data communications or with tourists. Unfortunately, an alarming portion

of the viruses spread through infected original disks. These are usually the result of sloppy virus control by either the original manufacturer of the program or the resellers facilities.

The virus situation in Scandinavia is getting constantly worse. Computer hobbyists with interests towards viruses are adapting manners from their foreign colleagues. Virus-artists use cool aliases or handles, speak their own language and try to gain big reputations. Several underground bulletin board systems (BBSs) focusing on viruses have been established.

Nordic virus authors have begun to organize themselves as groups. Several international virus writing groups are known, such as Phalcon/Skism, Rabid, Youngster against McAfee, NuKE and Crazysoft. Similar groups in Scandinavia are Demoralized Youth and BetaBoys.

The group Demoralized Youth is lead by Tormentor from Göteborg, Sweden. He is well known for the several advanced viruses he has written. Demoralized Youth also runs its own BBS, The Swedish Virus Laboratory, where members of the group discuss their newest achievements and exchange viruses. The group, which has members in Sweden and Norway and is seeking new members from Finland, has released about ten viruses.

The BetaBoys group has released a series of simple but very destructive viruses. These releases are made through official boards, which carry the questionable honour of being an official BetaBoys distribution site.

Several virus boards advertise themselves quite openly, offering viruses to practically anybody. Viruses are also sold. The almost astoundingly open manner in which virus programmers distribute their creations is made possible by outdated laws that do not recognize computer viruses. As unbelievable as it sounds, creating and spreading intentionally destructive programs is not illegal except in few rare cases.

All in all, at least 20 viruses have been written in Sweden, around 10 in Iceland, 5 in Denmark, at least three in Norway and over 15 in Finland. Some of these viruses are very complicated and obviously written by experienced programmers. Several Scandinavian viruses have managed to spread around the world, even to USA and Asia.

Epidemics caused by viruses are typically local. Usually a virus arrives in an organisation via a floppy obtained from an outside source. The virus starts to spread from one computer to another as users exchange data and through the file server when users are given write access to program areas in the file server. After a varying time, the virus activates on some or all the workstations and gets noticed.

The actual number of virus occurrences and the extent of the resulting damages cannot be estimated accurately since, unfortunately, most virus incidents are not reported due to fears of adverse publicity to the organisation in question. In several known cases the damages have been significant, though.

Additional information on different viruses can be obtained through the **Information** item in the **Viruses** menu.

11.3 Working Mechanisms of Viruses in MS-DOS

The following section presents some of the most common virus types and their working mechanisms. There are four major types of viruses:

- file viruses
- boot sector viruses
- companion viruses
- cluster or link viruses

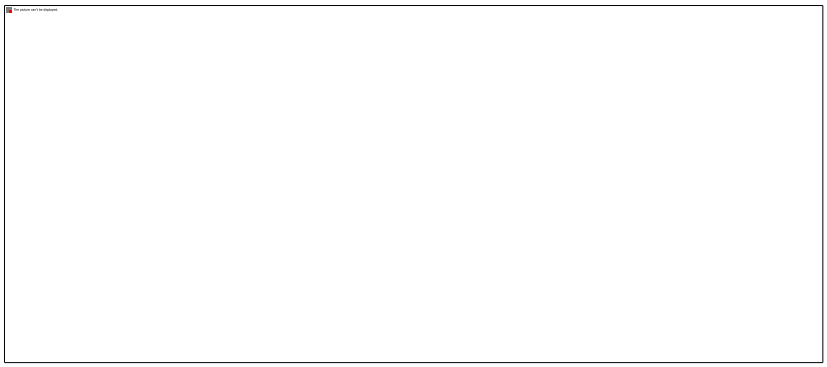
FILE VIRUSES

File viruses infect executable programs, usually COM and EXE files, but sometimes also overlay files. An overlay file may have any extension, but the most common ones are OVL and OVR. These files have one common aspect, they contain executable code.

Some viruses infect files of many types. For instance, the Amoeba virus infects both COM and EXE files.

A file virus functions by finding a suitable victim, an executable program, into which it adds its own code.

A virus typically adds its code to the end of the host file. To make sure that it will be executed with the parent program, the virus adds a jump instruction to the beginning of the program. The jump transfers control to the viral code at the end of the host code.



After being activated, a virus has free reign in the computer. Normally the primary function of a virus is to spread the infection. This can be done in different ways.

When an infected program is run, the virus may stay resident in memory and infect every program executed. Viruses that use this method to spread the infection are called Resident Viruses.

Other viruses may search for a new file to infect when activated. After infecting a specified number of new files, the virus transfers control to the original program. Viruses that use this method to spread the infection are called Direct Action Viruses.

After control is returned to the original application its execution will continue as if nothing had happened.

After the infection, the virus may move into its action phase, intended to attain a specific goal set by the virus writer. The transition may happen on a specific date or when some condition has been met. In its action phase, a virus may destroy data or perform some other disruptive action.

A "time bomb" like this is often designed to be harmful, maybe doing something like formatting the hard disk. Sometimes it is relatively harmless, perhaps slowing the computer down every Friday or making a ball bounce around the screen. However, there is not really any such thing as a harmless virus. Even if a virus is not intended to cause damage, it may do so in certain cases, often due to the incompetence of its author. Programs that do a self-check on themselves will usually refuse to run after infection. A virus may be modified, either by the original author or someone else, so that a more harmful version of it appears. It is also possible that the

modification produces a less harmful virus, but this happens only rarely.

The damage caused by a virus may consist of the deletion of data or programs, maybe even overwriting or reformatting of the hard disk, but more subtle forms of damage are also possible. Some viruses may modify data or introduce typing errors into text. Other viruses may have no intentional effects other than just replicating.

The actions of a virus will slow down the start of the actual program. The delay is often so slight that it is almost impossible to notice. There are, however, a few viruses that infect a large number of files at once after being executed. This may slow down the start of the actual program by tens of seconds.

When a virus adds its code to the beginning or end of a program, the size of the file will change. Some viruses avert this by using an unused area in the host file to store their code. This way the actual size of the infected program does not change.

Most viruses try to recognize existing infections to avoid reinfecting programs already infected. Usually viruses mark infected files with some sort of easily distinguishable marker, such as setting the time stamp of the file to an "impossible" value. This makes it possible to inoculate programs against specific viruses, by making the "victim" appear to be infected. However, this method is useless as a general defense, as it is not feasible to inoculate a program against multiple viruses.

BOOT SECTOR VIRUSES

A boot sector virus infects the boot sector on a hard disk or a diskette. Normally the boot sector contains code for loading the operating system files. A boot sector virus replaces the original boot sector with a copy of itself and stores the original boot sector somewhere else, or simply replaces it totally. When a computer is later booted from such a diskette, the virus takes control and hides in RAM. It will then load and execute the original boot sector, and everything will appear to be normal. Except, of course, that practically every diskette inserted in the computer will be infected with the virus.

A very important fact to realize is that every formatted diskette has a boot sector. Every diskette (both system diskettes and data diskettes, diskettes with programs on them and diskettes with no files at all) has code in its boot sector. This code is run when a computer is booted from the diskette. If the diskette does not contain the operating system, the program code in the boot sector will print a message, such as:

```
Non-system disk or disk error
```

If the diskette is infected with a virus, it will now be active, and has usually already infected the hard disk. If the aforementioned happens, turn the computer off, or press the reset button. Then boot the machine from a clean diskette and check the hard disk for viruses. It is important to note that pressing Ctrl-Alt-Del will not be sufficient, as some viruses can survive that.

The best way to protect a computer against boot sector viruses is to prevent booting from diskettes. This can be easily done with the BIOS SETUP of the computer. In older models this option may not be available.

If a computer has no hard disk, and is booted from a diskette, you should always use the same diskette, and keep it write-protected.

Most boot sector viruses infect main boot records on hard disks. This causes some problems, since main boot records cannot be cleaned with the DOS FORMAT command. The best way to disinfect main boot records is to use F-PROT. The other option is to low-level format the disk, repartition it with FDISK, format it with FORMAT and restore the contents from a backup. It is also possible to use FDISK's undocumented /MBR switch, but even it doesn't necessarily be able to clean the virus.

Because the size of a boot record is limited to 512 bytes on diskettes, large boot sector viruses have to hide part of their code outside the boot sector itself. Many viruses create bad sectors on the disk and store their code on those sectors. It is also possible to hide code in the area reserved for the main directory. A further option is to format an extra track and use it for storing the virus code. Only reasonably sophisticated viruses use the last option.

As there are five different diskette formats available (360 KB, 1.2 MB, 720 KB, 1.44 MB and 2.88 MB), only few viruses can infect all diskette types.

A boot sector virus will usually hide at the top of memory, thus reducing the amount of memory that the DOS sees. For example, a computer with 640K might appear to have only 639K. It should be noted that some BIOS models also take one or two kilobytes away from the DOS memory.

F-PROT searches for boot sector viruses by checking the Master Boot Record (MBR), the boot records, the partition table and other available hiding spaces.

Remember: Never boot a computer with a hard disk from a diskette because it is practically the only way the hard disk can become infected with a boot sector virus.

Companion Viruses

Companion viruses use a feature of DOS to force their execution. When several files with the same basename but a different extension are in the same directory, the one with the COM-extension is executed first. This happens only when the command line entry does not include the extension.

Companion viruses use this feature by selecting an EXE file and creating a COM file with a similar name in the same directory. The COM file may be hidden and thus would not show up in a directory listing. It contains the virus code, of course.

It is also possible to design a companion virus that takes advantage of the order of directories specified in the PATH environment variable, or a virus that would use some other way to get its code to be executed before the host program.

After being activated, such virus executes the actual program and seizes control again after the execution of the program is ended.

F-PROT recognizes all known companion viruses.

Cluster Viruses

The first cluster virus was found in the end of 1991. This virus is called DIR-II and it is known as one of the fastest spreading viruses in existence.

A cluster virus does not make any changes to the code of the program it infects. Instead, the virus changes the information for the program file in the file system's FAT area. It redirects the pointer for the actual program to point to its own code located somewhere else on the disk.

When an infected program is executed, DOS will load the virus code instead of the first cluster of the actual file.

DIR-II stores its own code in the last two clusters on a disk and reserves the clusters in the FAT. After the disk is completely infected, every entry in the FAT for an executable file points to the second-to-last cluster on disk. DIR-II saves the original address of the first cluster of a file into two unused bytes in the directory entry for the file. DIR-II uses this information to execute the original file.

DIR-II is a stealth virus and able to hide its actions when active. It infects files every time DOS reads a directory sector. Use of the DIR command, for instance, infects every file in the current directory, if DIR-II is active in memory.

F-PROT eliminates DIR-II by decrypting the original cluster addresses and repairing the FAT tables with this information.

11.4 Advanced Methods Used by Viruses

The following section presents some of the methods viruses use to hide their presence or to otherwise complicate the process of fighting against them.

The following advanced methods are described:

- the use of stealth techniques (stealth viruses)
- encryption and mutation techniques
- anti-anti-virus techniques (retroviruses)
- application specific techniques (application specific viruses)
- the use of several infection techniques in one virus(multipartition viruses)

STEALTH VIRUSES

Some years ago it seemed that the final weapon against virus infections had been invented. This omnipotent technique was called checksumming.

The checksummers calculate an individual signature for each file. By recalculating the signature and comparing it to the original one stored in a database, it is possible to detect every change made to a file. However, it did not take long for the first file-infecting stealth viruses to appear and crush the hopes of final victory in the battle against viruses.

A stealth virus falsifies the information read from a disk so that a program reading the disk receives incorrect data. The virus does this by intercepting the interrupt vectors used to read data from a disk and supplying the reading application with false information. This way the program reading the disk will receive information which incorrectly indicates everything to be all right. This technique can be successfully used by both file viruses and boot sector viruses.

A stealth virus does not attempt to break CRC-calculation routines. All that is needed is to supply the operating system with normal-looking information.

A good example is a virus called Brain, which was the first known stealth virus. Brain is a boot sector virus which transfers the original contents of the boot sector to a suitable location on the disk. It is also too big to fit completely on the boot record and needs to store a large part of its own code on the data area of the disk.

If a computer is booted from an infected diskette, the virus is executed first. After taking control, Brain executes the original boot

sector and everything will look normal to the user. Brain observes all disk reads and writes, however.

When an application attempts to read the contents of the boot sector Brain responds by delivering the original boot sector code from its hiding place on the disk. Since the application actually sees the original code, everything appears to be in order even though the diskette is infected. Brain also redirects all write attempts to the boot sector, protecting its own code. Similar stealth methods have also been used in file infecting viruses.

A stealth virus can intercept all disk reads to present false information. For this reason, F-PROT has to check RAM memory before execution. If a stealth virus were already active it could falsify all disk reads. Even a stealth virus cannot hide its code in memory completely and it is always possible to find an active virus by checking all available memory before starting a virus scan.

Self-Encrypting, Polymorphic and Mutation viruses

Most virus scanners operate by searching for virus signatures. By analysing an existing virus and isolating a small part specific to the virus, it is possible to recognize all infections. The only prerequisite for recognition is that the piece of virus code used for the signature can be found.

To make the use of signatures impossible many different types of techniques can be used. Common to all of these is that the virus code changes between infections.

Mutating viruses change their code and sometimes even their functionality between generations of the same virus.

The most common mutation technique is encryption. Many mutational viruses encrypt their code with a simple encryption algorithm that uses the time of day as the encryption key, for instance. This makes all generations of a single virus different from each other except for the decryption routine in the beginning of the viral code.

To make it impossible or inconvenient to use the decryption routine as a signature, virus writers often try to minimize the size of the routine. This is based on the maxim that very short signatures should not be used because the possibility of finding the same byte sequence in normal applications increases.

Most viruses encrypt themselves only at the time of infection, but there are some that do it while they are resident in memory, too. Whale is one of these very complex and sophisticated viruses.

Dark Avenger is a Bulgarian virus writer who has created a mutation engine which is called MtE or Dark Avenger's Mutation Engine.

The MtE can easily be incorporated into any virus. The result is a virus that is functionally similar to the original virus but is attached to the host program in an endless variety of different versions. The MtE is distributed through underground bulletin boards as an object file that can be linked to any old or new virus.

It is extremely hard to isolate a signature from an MtE-encrypted virus that could be found in the next generation of the same virus. The encryption algorithm used by MtE is so advanced that only a single common byte can be found in all MtE-encrypted files. Furthermore, the location of this byte varies. The common instruction is JNZ (Jump if Not Zero), which is commonly used and can be found in practically every program available.

The MtE uses many different encryption algorithms and randomly adds extra bytes to the decryption routines.

MtE is not the only know mutation generator. TPE, NED and DAME are similar products, and there are several polymorphic viruses that have been created without the help of an external generator.

There are other methods to generate mutating viruses besides encrypting their code. One is to build the virus from very small modules that can be swapped inside the code without any functional harm.

The methods used to find mutational viruses are called algorithmic methods. A large part of the latest viruses must be identified by algorithmic methods because signatures cannot be used due to the structure of the viruses.

RETROVIRUSES

The actions of some viruses are intentionally directed against known anti-virus applications. A virus may search for files from an anti-virus package and delete them. An active virus may identify the execution of an anti-virus product and simply crash the computer. As a result, the user may simply think that the virus scanner is buggy and causes the crashes.

More intelligently designed viruses do not aim for the destruction of an anti-virus product. It would, of course, be a sure sign of something strange taking place and consequently the virus would be found shortly. A more devious way to incapacitate an anti-virus application is to make certain changes to the application itself. The virus scanner would still appear to be working normally but it would not find any viruses.

The Peach virus is a good example of a retrovirus. It is a standard file virus but has some features that are directed against anti-virus software. When Peach infects EXE files, it checks a certain byte from the file's header information. If the byte has a certain value, the file will not be infected. Peach apparently is trying to verify whether the file is a certain known application. Virus experts have yet to find out which application Peach is trying to avoid.

After being executed 27 times Peach attacks Central Point Anti-Virus software if it is installed in the same computer. CPAV saves its search signatures in a single file on disk. Peach deletes this file. CPAV does not check for the existence of the signature file and can afterwards be used as if all were in order; however, it will not recognize any viruses, because it lacks the signature file.

Due to the existence of retroviruses F-PROT utilizes many techniques against tampering of F-PROT files.

Application Specific Viruses

Some viruses look for a certain application from the computers they have infected. Examples of this virus type are:

- Macho/Syslock — looks for the string "Microsoft" and replaces it with "Machosoft".
- Flip — searches for an unknown application with a signature. The unknown application is probably some anti-virus software.
- dBase — corrupts dBase files.
- AntiCAD — when the virus notes the execution of AutoCAD it overwrites all hard disks and diskettes. AntiCAD also scrambles the CMOS-memory.
- Haifa — looks for assembler files (ASM files) and adds a small piece of code to the beginning of these files. This code destroys the boot sectors of disks.

Multipartition Viruses

Multipartition viruses utilize multiple infection techniques. They can infect different types of executable files, boot sectors, main boot records, FATs and directories.

Multipartition viruses have a better chance of surviving a cleaning operation than viruses of other types. Even if the virus is disinfected from all program files, it will infect them again if it has not been removed from the boot sector.

Some multipartition viruses can infect files without the COM or EXE extension. To find such infections, one should check all files instead of just the normal executable ones. A virus may survive a simple scan this way.

Tequila is an example of a multipartition virus. It infects both EXE files and the main boot records. The virus features encryption mechanisms and advanced hiding abilities.

11.5 How Great is the Virus Threat?

Viruses are a serious threat to information security. If security is lax, a well-designed virus can spread almost unnoticeably from one computer to another in the course of time. Given enough time, a virus can infect virtually all computers and — even worse — back-ups in an organization. A virus can even corrupt data bit-by-bit. If the designer of such virus has been skillful (and devious) enough, the changes can be so minute that it is next to impossible to notice them for a considerable period of time. Even if small discrepancies in the data are found, they are often thought to be the result of operator errors or other human factors. This can be the worst sort of damage, as even the backups cannot be trusted after the virus is found.

After spreading itself during the latency period, a virus may activate and wreak havoc in the computers. The extent of such damage can range from the annoying to the truly devastating. Many companies can be left totally crippled if the files in their computers are wiped out. A serious virus attack can be a catastrophic experience for a company.

Viruses are set apart from most other information security risks by the fact that even regular back-ups aren't always a sufficient precaution. Making frequent back-ups diminishes the danger of a total disaster in case of a virus attack, but since even the back-up copies can be infected or corrupted, other measures are needed.

Even though the likelihood of a disastrous virus infection is small, the danger is real and needs to be acknowledged. The cost of developing appropriate information security guidelines and purchasing effective anti-virus software is very small compared to the possible cost of an uncontrolled virus infection.

The rest of this chapter presents three very different cases of virus infection and the lessons learned from them.

CASE: LEHIGH UNIVERSITY

The Lehigh University was ravaged by a virus infection in October 1987. The virus destroyed information in 500 DOS microcomputers and infected about 3000 diskettes.

The infection spread primarily through the microcomputers in the university's library, which were in public use, and through the program diskettes the library lent out. The victims included researchers, students and companies that had ties with the university.

The price of disinfection was apparently very high. Almost all the victims were highly educated and knew computers well.

The Lehigh virus would have little chance of spreading widely today. The case illustrates, however, that ignorance about viruses can lead to serious consequences.

Viruses have grown immensely more sophisticated since the days of the Lehigh virus, but the virus-awareness of users has not necessarily risen accordingly. In an organisation, where employees have no education about viruses, a modern equivalent of the Lehigh virus can cause serious damage. Knowledge and awareness are some of the most important weapons in the fight against viruses.

CASE: AIDS INFORMATION DISKETTE

In December 1989 thousands of microcomputer users received a letter containing a brochure and a diskette. Almost all of them inserted the diskette in their computers and executed the program.

The program was a Trojan Horse which installed a time bomb on the hard disk. After 15-90 days the Trojan Horse encrypted the hard disk contents and printed an ordering form on the computer's default printer. The form promised that for $378 the company which had made the program would de-encrypt the hard disk. Thousands of users were left to wonder at the tens of megabytes of illegible trash.

The success of the AIDS Trojan Horse in such a large scale is due to the gullibility of the users.

The AIDS case teaches an important lesson:

Be always suspicious. The fact that a program appears to be legitimate does not guarantee anything. The AIDS program, of course, is a blatant case of attempted blackmail. It is not the only one of its kind, though. A number of other Trojan horse programs have been discovered, some of which are potentially very destructive.

Diskettes or software of unknown or dubious origin should always be checked for viruses before trying them.

Even original, shrink-wrapped copies of major software packages are not always safe. In a few cases, well-known software vendors have shipped virus-infected programs. The risk is small, but even commercial software should not be trusted blindly.

A second, frightening conclusion can also be drawn: the AIDS case proves beyond doubt that some people are willing to use viruses or Trojan horse programs for personal gain.

CASE: CASCADE INFECTION IN FINLAND

In the spring of 1992, the local area network of a large Finnish corporation was infected by the Cascade virus. The company did not use anti-virus software, which was the main reason for the large number of infected computers.

The virus spread for weeks. When the infection was noticed, many file servers in the LAN were infected. Most of the workstations connected to the servers were infected as well.

The disinfection operation took weeks. All disks, diskettes and tapes had to be checked and disinfected. The cost of the operation was extremely high.

This case is a classic example of how a small investment could have averted a major disaster. Anti-virus software is the best — and only — insurance policy against virus attacks.

11.6 Viruses in OS/2

The OS/2 environment has so far been free from viruses written specifically for that operating system. No OS/2-specific virus has been found.

The reason for this is probably the small market share of OS/2 and the fact that OS/2 applications are more difficult to program than DOS applications. This applies especially to viruses which are mostly coded in assembly language.

Technically there are no reasons why an efficient OS/2 virus could not be written. In fact, OS/2 would provide viruses with much better services than DOS.

Even with no specific OS/2 viruses in existence, OS/2 is still not safe from the virus threat. Several normal DOS viruses work quite well in OS/2's DOS boxes.

Boot sector viruses can contaminate OS/2-machines and destroy information in them just as easily as in DOS computers. This is because the boot sector is loaded and the code it contains is executed before the operating system is loaded in the first place.In the case of OS/2 this means that the virus will not be functional after the operating system is finally read but it has ample time to do harm before that.

DOS Services

Several DOS viruses function normally in OS/2's DOS boxes. Everything might not work as expected, however. A boot sector virus may, for instance, infect OS/2's boot sector but will not work after the operating system is loaded. Normally, the system will not boot at all.

Normal file viruses may infect OS/2 applications as well, as long as they follow the DOS file naming convention. The most common result is that the infected application cannot be executed because the virus was not designed to infect program files that use the NE (New Executable) header.

Most viruses use DOSs interrupts. Because DOS's INTs do not function in OS/2 this makes OS/2s protected mode safe from these viruses. In DOS windows, DOS interrupts function normally and so do the viruses.

The HPFS Filing System

DOS viruses spread in HPFS based systems in the same way as they do in FAT systems. Boot sector viruses may corrupt the disk if they try to modify the FAT which can not be found under HPFS.

In HPFS, the boot sector is in the standard location. This makes it possible for a normal boot sector virus to infect HPFS boot sectors. If a virus infects OS/2's boot sector, OS/2 may not boot. If booting is still possible, the virus will not remain active as OS/2 overwrites the whole memory and resets the interrupts.

Boot sector viruses will not be able to spread from an OS/2 system because OS/2 does not support the INT 13h interrupt.

Under HPFS it is possible to use long file names — current DOS viruses will not infect OS/2 applications with names longer than the standard 8+3 characters.

The Boot Manager

The Boot Manager enables the user to install several operating systems on the same hard disk. It creates a one megabyte partition that can be manipulated from DOS or OS/2 only with direct disk read/writes. The partition is used by the Boot Manager.

When a computer with several operating systems is booted, the Boot Manager asks which operating system should be loaded. If the Boot Managers partition has been infected by a boot sector virus (FORM, for instance) the end result will depend on which operating system

the user decides to load. If OS/2 is loaded, the virus does not survive but if DOS is loaded the virus will activate normally.

REXX

OS/2 includes a powerful command language, REXX. With the REXX commands it is possible to design a destructive command line virus. Several such viruses have been created for test purposes but none have been discovered in the wild.

The threat generated by command line based viruses should not be considered equal to the threat of normal viruses. Command line viruses spread only in command line files which can easily be checked with an editor.

11.7 Viruses in Windows

It has taken surprisingly long for the first Windows specific viruses to appear. The main reason for this is the difficulty of Windows programming, which is much more demanding than DOS programming. In addition, the Windows file structure is more complex than that of normal DOS files. It will be, however, only a matter of time before virus programmers gain sufficient knowledge on Windows programming. The first Windows viruses have, in fact, already been found.

Windows is more vulnerable to viruses than OS/2 because the functional parts of a DOS virus do not have to be modified for the virus to work under Windows. Under OS/2, the code would have to be written practically from scratch.

The only parts of a DOS virus that have to be changed to work under Windows are the infection mechanism and activation routines.

The Structure of Windows Applications

Windows and DOS programs have a different internal structure.

Both begin with DOS's MZ header that enables DOS to read and execute the program. In Windows applications, the MZ header consists of a short reference to a routine that is run when the program is executed from DOS. *("This program requires Microsoft Windows")* The header also includes the disk address for the Windows header (NE header).

The Windows NE header (New Executable) is actually same as the one used for OS/2 applications. The NE header does not have a set location in a program file. Instead, its location is given in the MZ header.

Windows also supports other kinds of headers:

- LE - linear executable (for instance. *.386 device drivers)
- W3 - WIN386.EXE
- PE - portable executable (Win32 and NT)

A Windows application gives a virus better hiding places than DOS applications. One example is ordinary dynamic libraries that can be given any kind of file name extension. The files that can contain executable code (and viruses on the side, of course) can no longer be recognized by their extensions.

In addition to the ordinary EXE files, a virus can infect also font files or display drivers.

The Execution of an Application and Interrupts

Under Windows, the execution of an application is done by calling the WinExec()-function. The function generates the following interrupt call:

INT 21h AH=4BH (EXEC)

The same interrupt is used in DOS. An DOS application that is watching the interrupts does not see this interrupt, however. The following interrupt is seen instead:

INT 21h AH=3DH (OPEN)

What is the reason for this?

Windows is a DOS extender that functions in a protected mode. It uses Int 21h calls in protected mode. Due to the fact that DOS can not execute protected mode applications Windows's DOS extender (WIN386.EXE or DOSX.EXE) provides its own implementation of the EXEC. The EXEC of the extender uses DOS file services to read an application to memory.

Kernel

The Windows environment is at its most vulnerable during the bootstrap process. The Windows kernel does not have its own file management functions and uses DOS services (Int 21h).

The kernel manages the Windows bootstrap even if it has an NE header. The process is as follows:

- The execution of the stub code that is located in the MZ-header
- The application file is opened (the kernel itself)
- The NE address is retrieved from the MZ header
- The correct start address is retrieved from the NE header (CS:IP)
- The value 'OK' (4B4Fh) is written to the AX register

- The CS:IP is pushed to the stack and a RETF is executed
- BOOTSTRAP is executed
- BOOTSTRAP executes the DPMI services
- The old interrupt vectors are saved along with the DPMI exception vectors. The vectors are replaced with new ones that Windows will use.
- The rest of the Windows system is loaded

THE FUNCTIONING MECHANISMS OF VIRUSES IN WINDOWS

It has taken a long time for the first Windows specific viruses to be written. There have, however, been several DOS viruses that infect Windows applications. The most probable result of such an infection is that the Windows application can not be executed any more.

The reason for this is that the virus writer has not taken into account the possibility of an abnormal header. Some of the more advanced viruses have been able to infect the DOS stub. The execution of a Windows application infected with such a virus results in the activation of the virus. This works only when an attempt has been made to execute the Windows application from DOS.

Even a virus of this kind would not spread under Windows. It would function only in DOS.

The writing of a direct action virus for Windows is technically more challenging than writing a similar DOS virus. Unfortunately, a Windows virus can be built by modifying the infection mechanism of an existing DOS virus. The structure of the NE header will have to be taken into account, of course, which makes the programming of a Windows virus a little more difficult. The first virus like this was found in September 1992 (Win_Vir.14).

The programming of a Windows virus functioning as background process (the equivalent of a DOS TSR) is much more difficult. The Windows memory protection scheme makes it harder for programs to modify memory areas reserved for other applications. Protections like this do not exist in DOS at the moment.

Memory protection gives no shelter against viruses because very few virus writers have used techniques like this even in DOS.

A special development toolkit exists for the designing of device drivers. DDK (Device Driver Kit) gives the programmer free reign in the computer with no limitations. The most dangerous Windows viruses will probably be written with the DDK.

The use of OLE (Object Linking and Embedding) generates interesting possibilities on virus infection. With OLE it is possible to include

executable code in documents. This means that we will some day have to check documents for viruses as well as programs.

WIN_VIR.14

Data Fellows Ltd analysed the first Windows specific computer virus in the fall of 1992. This virus recognizes the Windows NE files and uses direct action methods against Windows applications. The virus does not infect normal DOS applications.

The virus sample was received from Sweden in the September, 1992. The results of preliminary analysis are as follows:

- The virus infects only Windows EXE files
- The strings 'Virus_for_Windows v1.4' and 'MK92' are embedded in the code
- The virus infects only Windows applications. The infections are generated at the moment of executing an infected application.
- As a result of the infection mechanisms used by the virus an infected file does not start with first double click but only with the second.

The virus does not constitute a major threat to Windows users. It is not a very efficient infector and does not try to harm data.

The infection procedure:

1. The virus is activated when an infected application is executed.
2. The virus searches for a file suitable for infection from the default directory using DOS's Int 21h, AX=4E, 4F services
3. If no targets can be found, the execution is finished with the call Int 21h, AX=4C00h. The actual Windows application is not executed.
4. If targets are found, they are opened one by one and the time stamps saved in memory.
5. The MZ and NE headers are checked.
6. Several values are checked from the NE header.
7. The virus code is added in the middle of the application.
8. The replaced code is moved to the end of the application.
9. The CS:IP from the NE header is changed to point to the beginning of the viral code.
10. The virus deletes its code from the original file and rebuilds it to a functional state,
11. The execution is finished.

Other observations:

- After the virus code is executed, the original application is not executed. This will seem as a failed double click. As the virus rebuilds the original file if it manages to infect a new file, the next attempt to execute the original application is successful.
- The infected files grow with 854 bytes.
- The infection does not change the time stamp of the target application file.
- The virus is not encrypted or protected in any way.
- No activation routines could be found.
- The name of the infector application and the name of the infected file is saved in the virus code.

12 How to Use Your Computer Safely

- Never boot your computer from an unknown diskette.
- If you forget a diskette in a drive and start the computer, remove the diskette, turn off the computer using the power switch and restart. If you have the slightest reason to suspect that there is a virus on the diskette, boot your computer from a clean system diskette and check your computer with F-PROT. A boot sector virus can infect your hard disk immediately if you boot your computer from an infected diskette.
- Create a plan for making back-up copies and stick to the plan. Keep several generations of back-ups. Keep both full backups and backups of just data files.
- Keep your diskettes write-protected whenever possible
- Don't use illegally copied programs or programs of uncertain origin. If you dont know the origin of a diskette, dont try running the programs on it, before checking the diskette for viruses.
- Check also the purity of original diskettes. They may be infected, too.
- Don't copy programs from one computer to another. Always install programs from original distribution diskettes or write-protected back-up copies.
- Install VIRSTOP and keep it active at all times
- Scan your hard disk regularly with F-PROT
- Dont underestimate the virus threat. Viruses are a major hazard.

Glossary

Anti-Virus Diskette

- A diskette in which are stored programs and files needed to remove a virus from the computer and recuperate from the damages caused by it. This kind of a diskette is also called a panic diskette.

AUTOEXEC.BAT

- A command string file that is automatically executed when a DOS computer is booted. AUTOEXEC.BAT contains, among other things, the run commands for automatically executable files and parameters for environment variables.

BIOS (Basic Input-Output System)

- The part of an operating system responsible for its basic functions. BIOS is stored in the ROM memory. BIOS code is the first code to be executed when the machine is turned on.

Booting

- The startup of a computer. During startup the computer reads the SETUP information after which it loads the operating system into memory from either a diskette or a hard disk. The programs defined in the files AUTOEXEC.BAT and CONFIG.SYS are also executed.

Boot Manager

- A program that enables the use of several operating systems on the same computer, though only one at a time.

Boot Sector (Boot Record)

- An area located on the first track of diskettes and logical disks. Boot sector information enables the computer to read an operating system like, for example, MS-DOS.

Boot Sector Virus

- A virus that hides its code in the boot sector of a disk or a diskette and which is therefore automatically executed when the computer is turned on. Most boot sector viruses stay active in memory after their execution.

Call

- (Here:) A request for the operating system to perform a specified function.

Cluster Virus

- A virus that changes the FAT area's program file information so that all the addresses point to the viral code's address on the disk instead of the programs' first clusters. Thus the virus is always executed first regardless of the program the user is trying to run.

CMOS

- The battery-powered memory of PC-computers. The size of this memory is typically within the 32-50 byte range. CMOS contains information about, for example, disks, date and peripherals. It is not emptied when the computer is turned off, provided the battery still has power.

Companion Virus

- A virus that infects programs by creating a new file that is executed insted of of the original one. This new file contains viral code which, after having been executed, passes the control to the original program. According to the functioning method, companion viruses are called COM-EXE viruses, stand-in-place -viruses or path viruses.

CONFIG.SYS

- A text file that is automatically read when the computer is booted. Contains the run commands for device drivers and parameters for the hardware.

CRC, Cyclic Redundancy Check

- A technique in which a CRC program calculates a checksum for each inspected file. This checksum is used as a template during later inspection to reveal changes the file has undergone. Such changes are indicators of a possible, but not certain, virus presence. Some legitimate applications can, also, change their own code.

Device Driver

- A program that controls devices, such as the hard disk, the mouse and the like. The startup commands for device drivers are usually stored in the CONFIG.SYS file for automatic execution during computer booting. See also: Low-level Driver.

Disinfection

- An F-PROT function which cleans files, boot sectors and the like of viral code. After disinfection the functioning of cleaned objects is restored. If viruses have corrupted their victims' data during infection, disinfection is no longer possible.

DOS (Disk Operating System)

- An operating system generally used in microcomputers. DOS is loaded into memory from a disk or a diskette. The most common DOS is the one made by Microsoft, MS-DOS. IBM's PC-DOS and Digital Research's DR-DOS are other common alternatives.

DPMI **(**Dos Protected Mode Interface)

- An interface designed to be used by DOS applications. Via DPMI DOS programs get access to linear memory space and a more advanced command base.

EMS, Expanded Memory Specification

- A parallel memory that succeeds DOS's 640 kilobyte limitation by opening memory windows to a larger memory area. For example, the EMS 4.0 opens four 16 kilobyte windows to a 32 megabyte expanded memory.

Exit Code (Errorlevel)

- A code string, usually a number or a number sequence a program transfers to the operating system when it terminates its execution. The exit code generally relays information about the program execution and the termination status (for example, whether the execution has terminated normally or not).

Extended Memory

- The part of RAM exceeding 1088 kilobytes. This memory space is addressed directly by suitable applications, unlike Expanded Memory, which uses the windowing technique. Also: Linear Memory.

FAT (File Allocation Table)

- The allocation table used with the DOS operating system. FAT contains information about the location of files on a disk and also of bad (unusable) sectors.

File Server

- A network-connected computer on which are located the applications and data shared by the network users.

File Virus

- A virus that attachs its code into files. The infected files are mostly COM or EXE files, in other words executable programs.

F-TEST

- A part of the F-PROT program package. F-TEST is used to confirm the installation and active status of VIRSTOP.

Header

- (Here:) Information located at the beginning of an application, on the basis of which it can be executed in the environment it is designed for. The header type varies according to the environment (for example, DOS programs have MZ-headers, Windows and OS/2 applications have NE-headers et cetera.).

Heuristic Analysis

- One of the virus scanning methods. A Heuristic Analysis search is based on identifying the functional characteristics of viruses instead of identification by their "signatures" among normal program code.

High-level Formatting

- Logical formatting of a disk. For example, the FORMAT command performs a logical formatting on the hard disk, not actually overwriting the data but marking all areas as available.

High Memory Area, HMA

- A memory area of 64 kilobytes, located between 1024 and 1088 kilobytes in memory.

HPFS (High Performance File System)

- A file management system that can be used under the OS/2 operating system.

Icon

- A symbol in the form of a picture used in graphical user interfaces such as Windows. An icon is located on the screen, and when it is clicked with mouse the application or document it symbolizes is activated.

Infection

- The act of viral code placing a copy of itself in its victim, which can be a file, a program, a boot sector et cetera.

Interrupt Call

- A request made by the operating system to the hardware.

Kernel

- (Here:) The Windows core, which loads other parts of the Windows into memory during booting sequence.

Latency Period, Latency Phase

- A phase between the initial infection and the activation of a virus. During this period the virus does nothing else but infects more victims. See:Trigger.

LE-header (Linear Executable)

- A header type supported by Windows.

Linear Memory

- See: Extended Memory

Low-level Driver

- A device driver that is loaded into memory first. These device drivers take care of some of the operating system's functions, and all other device-controlling programs use their services.

Low-level Formatting

- The physical formatting of disks, which overwrites all memory spaces on a disk and marks them as available. Low-level formatting cannot be done on some hard disks.

Main Boot Record (MBR, Partition Boot Record)

- An area located on the zero track of physical hard disks. It contains the main boot program and the partition table. Main boot record is independent of operating systems and is always executed first after the computer has performed the Power-On-Self-Test (POST). The size of a main boot record is 512 bytes. The main boot program translates the partition table, which contains information on how the physical disk is divided (partitioned) into logical entities. After this, the main boot program executes the boot sector of the active partition.

Multipartition Virus

- A virus capable of infecting various targets. For example files, boot sectors and directories.

Mutation Engine (MtE, DAME)

- A program created by Dark Avenger, a Bulgarian virus writer. The program mutates viruses without affecting their functioning. Mutation Engine can be linked to any virus as a normal object file.

MZ-header

- The header type used in DOS files. It is used also in Windows applications, where it identifies the file as a Windows application and gives the disk address of the corresponding Windows header (NE, New Executable)

NE-header (New Executable)

- The header type used in Windows and OS/2 applications. When used with Windows applications, this header is addressed via a normal DOS header (MZ-header).

OLE (Object Linking And Embedding)

- A Windows implementation that makes it possible to include executable code inside documents.

OS/2

- A multitasking operating system with a graphical interface. Manufactured by IBM.

Packed File

- A file compressed to an archive form by an application designed for the purpose, such as ARJ or PKZip. As packed files cannot be directly read or executed in the archive form, they must first be extracted (= melted, expanded) to their normal form in order to perform operations on them them.

Packed Program File

- An executable file packed (compressed) with such applications as LZEXE, PKLITE, EXEPACK, DIET and ICE. These programs look like normal executable files, with the exception that their size has been reduced. When they are executed, they automatically extract (expand to normal size and form) and execute themselves.

Panic Diskette

- See: Anti-Virus Diskette

Partition Table

- Part of the main boot record of a hard disk. Contains information on how the physical disk is divided (partitioned) into logical entities.

PATH

- An environment variable that contains the directories that are searched when an execution command is given.

PE-header (Portable Executable)

- A header type supported by Windows.

POST (Power-On-Self-Test)

- A test performed at the initial stages of booting. During POST the computer checks the integrity of memory circuits and the existence of peripherals, after which BIOS loads the operating system.

Quick Scan

- The fast-scanning function of F-PROT which searches for viruses using only one signature per virus as a template. Quick Scan does

not enable complete virus identifications or program disinfection.

REXX

- The command language of the OS/2 operating system.

Secure Scan

- F-PROT's complete virus search which uses algorithms in addition to two or more indentification-template signatures for each virus to scan for infections. Full virus identification and disinfection are possible when using Secure Scan.

Signature

- A character string belonging to a virus.

Stealth Virus

- A virus that hides itself by intercepting the operating system's disk read calls and transmitting an image of clean, uncorrupted data to the disk-scanning program.

TridenT Polymerphic Engine, TPE

- An object library that can be linked to any virus, much like the infamous Mutation Engine. Its function is also the same; to encrypt and mutate viruses so that they are no longer recognizable. The functioning mechanisms of TPE closely resemble those of the Mutation Engine, but its encryption method is wholly different. TPE is believed to be the brainchild of Masud Khafir of the *TridenT Virus Research Group*.

Trigger

- An impulse that causes a virus or a Trojan Horse to move from latency to its active phase. In the active phase the malware starts to affect the functioning of the host computer by, for example, formatting the hard disk, writing messages on the screen et cetera. Almost anything the virus author can think up may act as the triggering impulse. It can be, for example, the current date, time lapsed from the initial infection, a character string in the input stream et cetera.

Trojan Horse

- A program in which a trap is hidden inside normal program code. These traps are sprung by triggers defined beforehand and perform functions that, from a user's point of wiew, are usually unpleasant.

TSR (Terminate and Stay Resident)

- A program that after its execution stays active in memory as a background program.

UMB, Upper Memory Block

- A memory area located between 640 and 1024 kilobytes in memory.

Vector

- A memory address that contains the pointer to the real physical memory address. Vectors are used in the interrupt handler of DOS operating systems. Interrupt routines are called via the vector table located in the lowest addresses of memory. The benefit of a vector table is compatibility between different versions of an operating system.

VIRSTOP

- The memory resident portion of F-PROT, which prevents viral activation and infections in the computer.

Virus

- A program capable of replicating itself. Besides multiplying its code, a virus usually tries to affect changes in other programs, files and in the functioning of its host computer.

Windows

- A graphical user interface made by Microsoft. Windows is not an operating system of its own, but operates over DOS.

Write Protection

- A protection on diskettes that prevents writing on them.

W3-header (WIN386.EXE)

- A header type supported by Windows.

Index

Additional Bytes; 27
Anti-virus diskette; 29; 31; 50
Application specific viruses; 60; 63
AUTOEXEC.BAT; 6; 7; 12; 13; 17; 18; 19; 30; 34; 41; 44; 50
Boot records; 29; 58
Boot sector viruses; 23; 28; 31; 43; 52; 55; 57; 60; 66; 72
Boot sectors; 28; 44; 52; 57; 63; 66; 72
Cluster viruses; 55; 59
Companion viruses; 55; 59
CONFIG.SYS; 7; 13; 17; 34; 44
ENGLISH.TX0; 9; 30
Extra track; 58
F-PROT.EXE; 6; 9; 12; 29; 41
F-TEST; 32
F-TEST.COM; 9; 32
False positives; 40
File servers; 9; 12; 13; 18; 21; 22; 42; 53; 55
File viruses; 23; 27; 43; 55
FPSETUP; 19
Heuristic Analysis; 21; 38; 39
HPFS (High Performance File System); 44; 66
Install; 6; 10; 11; 12; 13; 18; 30; 31; 44; 67; 72
Latency period; 52; 64
Local Area Networks (LANs); 9; 12; 13; 18; 19; 21; 33; 34; 42; 53; 65
Low-level Drivers; 18
Main Boot Records; 28; 29; 58; 63
Multipartition viruses; 60; 63
Mutating viruses; 60; 61; 62
Mutation Engine; 61
OS/2; 44; 65; 66
overlay; 55
Overlay files; 24
Packed files; 24
Partition tables; 58
Print server; 53
Program disinfection; 8; 20; 22; 27; 28; 31; 42; 50; 58
Quick Scan; 20; 42

Registration; 19; 33
Retroviruses; 60; 62
Secure Scan; 7; 8; 20; 30; 31; 42
Self-encrypting viruses; 60; 61
SETUP.F2; 9; 12; 13; 18; 19; 30
SIGN.DEF; 9; 30
Signature; 9; 10; 11; 17; 20; 23; 31; 32; 33; 38; 43; 61; 62; 63
Stealth viruses; 31; 60
Trigger; 51; 52
Trojan Horse; 52
Trojan Horses; 23
TSR; 9; 31; 34; 69
USER.DEF; 9; 24
VIR-HELP; 12; 30
VIR-HELP.ENG; 9
Viral distribution; 52; 53; 54; 58
VIRSTOP; 6; 7; 10; 12; 13; 17; 18; 30; 31; 32; 34; 44; 48
VIRSTOP.EXE; 7; 9; 10; 12; 13; 17; 30; 31
Virus descriptions; 9; 25; 27; 49; 55
Windows; 9; 32; 33; 34; 44; 67; 68; 69; 70
Windows viruses; 67; 68; 69; 70

Appendix 1: F-SCHEDULER

Appendix 2: F-CHECK

Appendix 3: Version Bulletins